Meta-Learning

Pavel Brazdil,
LIAAD-INESC Porto L.A./Faculdade de Economia, University of Porto, Portugal

Ricardo Vilalta
Department of Computer Science, University of Houston, USA

Christophe Giraud-Carrier
Department of Computer Science, Brigham Young University, USA

Carlos Soares
LIAAD-INESC Porto L.A./Faculdade de Economia, University of Porto, Portugal

Synonyms

Selection of Algorithms, Ranking learning methods; Hyper-parameter Optimization; Adaptive learning; Self-adaptive systems; Dynamic selection of bias; Learning to learn.

Definition

Meta-learning allows machine learning systems to benefit from their repetitive application. If a learning system fails to perform efficiently, one would expect the learning mechanism itself to adapt in case the same task is presented again. Metalearning differs from base-learning in the scope of the level of adaptation; whereas learning at the base-level is focused on accumulating experience on a specific task (e.g., credit rating, medical diagnosis, mine-rock discrimination, fraud detection, etc.), learning at the metalevel is concerned with accumulating experience on the performance of multiple applications of a learning system.

Briefly stated, the field of metalearning exploits the relation between tasks or domains, and learning algorithms. Rather than starting afresh on each new task, metalearning facilitates evaluation and comparison of learning algorithms on many different previous tasks, establishes benefits and disadvantages, and then recommends the learning algorithm, or combination of algorithms, that maximizes some utility function on the new task. This problem can be seen as an instance of the algorithm selection task (Rice 1976) [].

The utility or usefulness of a given learning algorithm is often determined through a mapping between a characterization of the task and the algorithm's estimated performance (Brazdil & Henery, 1994) [2]. In general, metalearning can recommend more than one algorithm. Typically, the number of recommended algorithms is significantly smaller than the number of all possible (available) algorithms (Brazdil et al., 2009) [].

Motivation and Background

The application of machine learning systems to classification and regression tasks has become a standard, not only in research but also in commerce and industry (e.g., finance, medicine, and engineering). However, most successful applications are custom-designed, the result of skillful use of human expertise. This is due, in part, to the large, ever increasing number of available machine learning systems, their relative complexity, and the lack of systematic methods for discriminating among them. The problem is further compounded by the fact that, in Knowledge Discovery from Databases, each operational phase (e.g., pre-processing, model generation) may involve a choice among various possible alternatives (e.g., progressive vs. random sampling, neural network vs. decision tree learning), as observed by Bernstein et al. (2005) [].

Current data mining systems are only as powerful as their users. These tools provide multiple algorithms within a single system, but the selection and combination of these algorithms must be performed before the system is invoked, generally by an expert user. For some researchers, the choice of learning and data transformation algorithms should be fully automated if machine learning systems are to be of any use to non-specialists. Others claim that full automation of the data mining process is not within the reach of current technology. An intermediate solution is the design of assistant systems aimed at helping to select the right learning algorithm(s). Whatever the proposed solution, there seems to be an implicit agreement that metaknowledge should be integrated seamlessly into the data mining system. Metalearning focuses on the design and application of learning algorithms to acquire and use metaknowledge to assist machine learning users with the process of model selection. A general framework for this purpose, together with a survey of approaches, is in (Smith-Miles, 2008) [].

Metalearning is often seen as a way of redefining the space of inductive hypotheses searched by the learning algorithm(s). This issue is related to the idea of search bias, that is, search factors that affect the definition or selection of inductive hypotheses (Mitchell, 1997) []. In this sense, metalearning studies how to choose the right bias dynamically, and thus differs from base-level learning, where the bias is fixed or user-parameterized. Metalearning can also be viewed as an important feature of self-adaptive systems, that is, learning systems that increase in efficiency through experience (Vilalta & Drissi, 2002) [].

Structure of the Metalearning System

A metalearning system is essentially composed of two parts. One part is concerned with the acquisition of metaknowledge from machine learning systems. The other part is concerned with the application of metaknowledge to new problems with the objective of identifying an optimal learning algorithm or technique. The latter part – application of metaknowledge – can be used to help select or adapt suitable machine learning algorithms. So, for instance, if we are dealing with a classification task, metaknowledge can be used to select a suitable classifier for the new problem. Once this has been done, one can train the classifier and apply it to some unclassified sample for the purpose of class prediction.

In the following sections we begin by describing scenarios corresponding to the case when metaknowledge has already been acquired. We then provide an explanation of how this knowledge is acquired.

Employing Metaknowledge to Select Machine Learning Algorithms

The aim of this section is to show that metaknowledge can be useful in many different settings. We will start by considering the problem of selecting suitable machine learning algorithms from a given set. The problem can be seen as a search problem. The search space includes the individual machine learning algorithms and the aim is to identify the best algorithm. This process can be divided into two separate phases. In the first phase the aim is to identify a suitable subset of machine learning algorithms based on an input dataset (Fig. 1 a-b). The selection method used in this process can exploit metaknowledge (Fig. 1 c). This is in general advantageous, as it often leads to better choices. In some work the result of this phase is represented in the form of a ranked subset of machine learning algorithms (Fig. 1 d). The subset of algorithms represents the reduced bias space. The ranking (i.e., ordering of different algorithms) represents the procedural search bias.

The second phase is used to search through the reduced space. Each option is evaluated using a given performance criteria (e.g., accuracy). Typically, cross-validation is used to identify the best learning algorithm (Fig. 1 e). We note that metaknowledge does not completely eliminate the need for the search process, but rather provides a more effective search. The search effectiveness depends on the quality of metaknowledge.

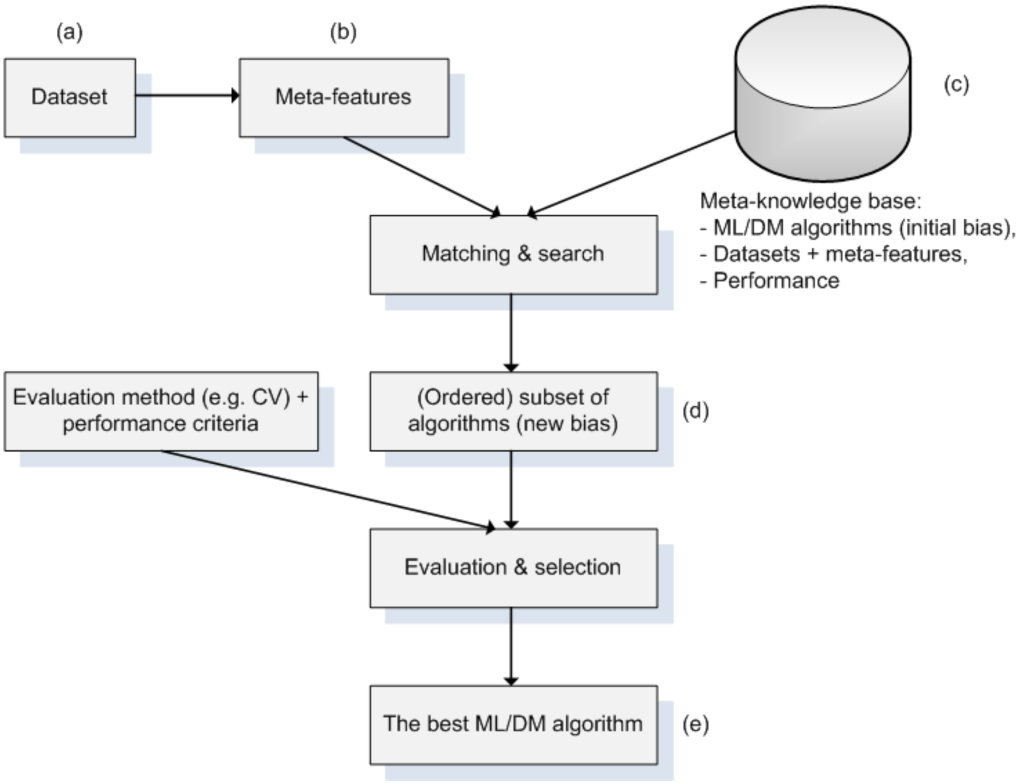
(a)
Dataset
(b)
Meta-features
(c)
Meta-knowledge base:
- ML/DM algorithms (initial bias),
- Datasets + meta-features,
- Performance
Matching & search
Evaluation method (e.g. CV) + performance criteria
(Ordered) subset of algorithms (new bias)
(d)
Evaluation & selection
The best ML/DM algorithm
(e)

Fig. 1. Selection of machine learning algorithms: Determining the reduced space and selecting the best alternative.

Input to and Output from the Metalearning System

A metalearning approach to solving the algorithm selection problem relies on dataset characteristics or metafeatures that provide some information to differentiate performance among a given set of learning algorithms. These include various types of measures, or meta-features, discussed in detail below.

Much previous work in dataset characterization has concentrated on extracting statistical and information-theoretic parameters estimated from the training set. Measures include the number of classes, the number of features, the ratio of examples to features, the degree of correlation between features and target concept, the average class entropy, etc. (Engels et al., 1998) []. The disadvantage of this approach is that there is a limit to how much information these meta-features can capture, given that all these measures are uni- or bi-lateral measures only (i.e., they capture relationships between two attributes only or one attribute and the class).

Another approach is based on what are called *landmarkers;* these are simple and fast learners (Pfahringer et al., 2000) []. The accuracy of these simplified algorithms is used to characterize a dataset and to identify areas where each type of learner can be regarded as an expert. An interesting variation on the theme of landmarking uses information obtained on simplified versions of the data (e.g., samples). Accuracy results on these samples serve to characterize individual datasets and are referred to as *sub-sampling landmarks*.

In principle, any machine learning algorithm can be used at the meta-level. However, one important aspect of the metalearning task is the scarcity of training data. As a result, many researchers in the past have used *lazy learning* methods, such as *k*-NN, since these delay the generalization of metadata to the application phase (Nakhaeizadeh et al., 1997[]). However, other types of models, such as neural networks, ranking trees, and bagging ensembles, have been proposed and proved rather successful (Sun et al., 2012[]; Sun et al., 2013[]).

There are several possible outputs or types of model a metalearning system can produce. Some focus on selecting the best algorithm in the set of available base learners; some attempt to predict the actual performance of individual algorithms; yet others assess the relative performance of different pairs of algorithms; finally some systems produce a complete ranking of the base learners, that can then be followed by minimal testing to identify the truly best algorithm for the user's dataset. One significant advantage of ranking methods is that they offer a next best alternative if the first algorithm seems to be sub-optimal. As the set of base learners may contain variants of the same algorithms, and it would be wasteful to test them all before moving on to other types of algorithms, a recent approach known as active testing has been proposed, which seeks to identify the most promising algorithm that has a chance of surpassing the best algorithm identified so far (Leite et al., 2012[]).

Acquisition of Metaknowledge

There are two natural ways in which metaknowledge can be acquired. One possibility is to rely on expert knowledge. Another possibility is to use an automatic procedure. We explore both alternatives briefly below.

One way of representing metaknowledge is in the form of rules that match domain (dataset) characteristics with machine learning algorithms. Such rules can be hand-crafted, taking into account theoretical results, human expertise, and empirical evidence. For example, in decision tree learning, a heuristic rule can be used to switch from univariate tests to linear tests if there is a need to construct non-orthogonal partitions over the input space. This method has serious disadvantages however. First, the resulting rule set is likely to be incomplete. Second, timely and accurate maintenance of the rule set as new machine learning algorithms become available is problematic. As a result, most research has focused on automatic methods.

One other way of acquiring metaknowledge relies on automatic experimentation. For this we need a pool of problems (datasets) and a set of machine learning algorithms that we wish to consider. Then we need to define the experimental method that determines which alternatives we should experiment with and in which order (see Fig. 2 for details).

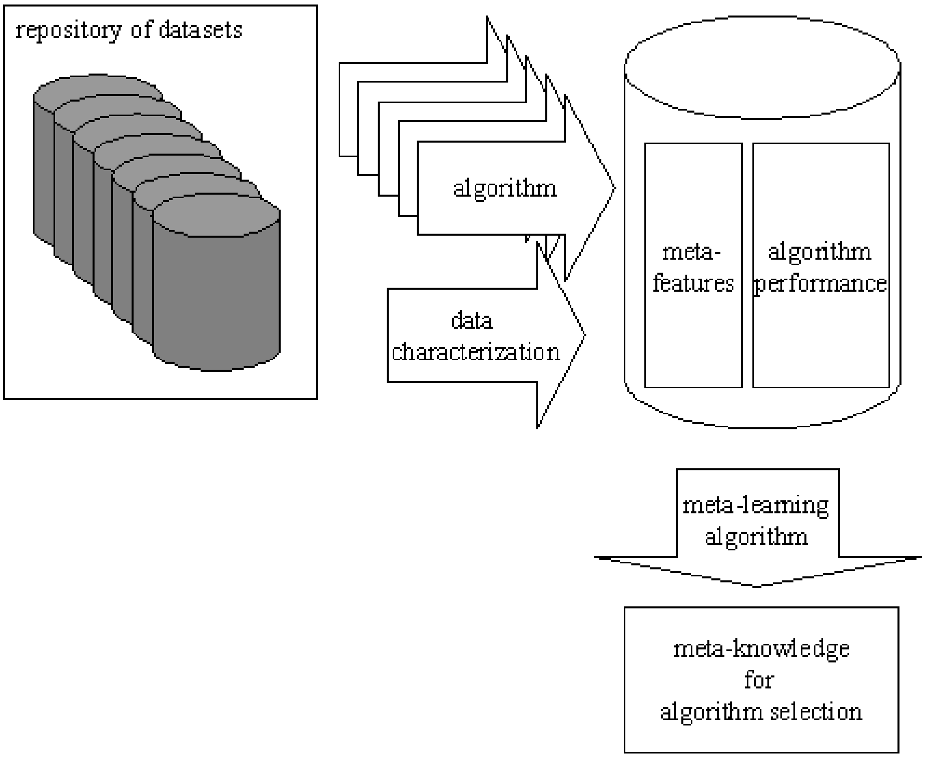

Fig. 2 Acquisition of Metadata for the Metaknowledge base

Suppose we have a dataset (characterized using certain meta-features), in combination with certain machine learning algorithms. The combination is assessed using an evaluation method (e.g., cross-validation) to produce performance results. The results, together with the characterization, represent a piece of metadata that is stored in the metaknowledge base. The process is then repeated for other combinations of datasets and algorithms.

Algorithm Selection and Hyperparameter Optimization

While this article describes metalearning in the context of selecting algorithms for machine learning, there are a number of other areas, such as regression, time series forecasting, and optimization (Smith-Miles 2008[]), where algorithm selection is important and could benefit from a similar approach.

Similarly, there has been recent interest in the optimization community in the problem of hyperparameter optimization, wherein one seeks a set of hyperparameters for a learning algorithm, usually with the goal of obtaining good generalization and consequently low loss (Xu, 2008[]). Hyperparameter optimization is clearly relevant to algorithm selection, since most learning algorithms have parameters that can be adjusted and whose values may affect the performance of the learner. Historically, metalearning has largely ignored parameter selection, and hyperparameter optimization has largely ignored metalearning. Recent efforts in bringing the two fields together hold promise.

Applying Metalearning to Workflow Design for KDD

Much of the work in metalearning has focused on classification algorithm selection, and thus addressed only a small fraction of the overall data mining process. In practice, users must not only select a classification learner but must often also consider various data pre-processing steps and other aspects of the process to build what are actually sequences of operations to apply to their data, also known as workflows. Several advances have been made in recent years in this area (Hilario, 2011[]; Kietz, 2012[]). Usually it is possible to distinguish two phases. In the first phase, the system runs different experiments that involve different workflows for many diverse problems. The workflow may be generated automatically with the recourse to a given ontology of operators. The individual problems are characterized and the performance of different workflows recorded. This can be compared to running experiments with a set of classification algorithms and gathering the meta-knowledge. In the second phase the system carries out planning with the aim of designing a workflow that is likely to achieve good results. In this phase, a given ontology of operators can again be exploited. The expansion of the operators may be guided by the existing meta-knowledge. The aim is to give preference to the more promising expansions and generate a ranked list of viable workflows.

Cross References:

Inductive Transfer

Recommended Readings

[1] Bernstein, A., Provost, F. and Hill, S. (2005). Toward Intelligent Assistance for a Data Mining Process: An Ontology-based Approach for Cost-sensitive Classification. IEEE Transactions on Knowledge and Data Engineering, **17**(4):503-518.

[2] Brazdil P. and Henery, R. (1994). Analysis of Results, in Michie, D., Spiegelhalter, D. J. and Taylor, C.C. (Eds.) *Machine Learning, Neural and Statistical Classification.* England: Ellis Horwood.

[3] Brazdil P., Giraud-Carrier, C., Soares, C. and Vilalta, R. (2009). *Metalearning – Applications to Data Minin*g, Springer.

[4] Engels, R. and Theusinger, C. (1998). Using a Data Metric for Offering Preprocessing Advice in Data-mining Applications. In *Proceedings of the 13th European Conference on Artificial Intelligence, 430-434.*

[5] Hilario, M. and Nguyen, P. and Do, H. and Woznica A. and Kalousis, A. (2011). Ontology-Based Meta-Mining of Knowledge Discovery Workflows. *In Meta-Learning in Computational Intelligence,* N. Jankowski et al. (eds).

[6] Kietz, J.U. and Serban, F. and Bernstein, A., and Fischer S. (2012). Designing KDD-Workflows via HTN-planning for Intelligent Discovery Assistance. In *Planning to Learn Workshop at ECAI-2012* (PlanLearn-2012)}, J. Vanschoren et al.(eds.).

[7] Leite, R., and Brazdil, P. and Vanschoren, J. (2012). Selecting classification algorithms with active testing. In *Machine Learning and Data Mining in Pattern Recognition*, 117-131. Springer.

[8] Mitchell, T. (1997). *Machine Learning,* McGraw Hill.

[9] Nakhaeizadeh, G. and Schnabl, A. (1997). Development of Multi-criteria Metrics for Evaluation of Data Mining Algorithms. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 37-42.

[10] Pfahringer, B., Bensusan, H. and Giraud-Carrier, C. (2000). Meta-learning by Landmarking Various Learning Algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, 743-750.

[11] Smith-Miles, K.A. (2008). Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. *ACM Computing Surveys*, **41**(1):6.

[12] Rice, J.R. (1976). The Algorithm Selection Problem. *Advances in Computers*, **15**:65-118.

[13] Sun, Q. and Pfahringer, B. (2012). Bagging Ensemble Selection for Regression. *In Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence*, 695-706.

[14] Sun, Q. and Pfahringer, B. (2013). Pairwise Meta-Rules for Better Meta-learning-based Algorithm Ranking. *Machine Learning*, **93**(1):141--161.

[15] Vilalta, R. and Drissi, Y. (2002). A Perspective View and Survey of Metalearning. *Artificial Intelligence Review*, **18**(2):77-95.

[16] Xu, L. and Hutter, F. and Hoos, H. and Leyton-Brown, K. (2008). Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. *Journal of Artificial Intelligence Research*, **32**:565--606.