International Journal of Computer Science & Applications
Vol. I, No. 1, pp. 31 - 45

Using Meta-Learning to Support Data Mining

Ricardo Vilalta[1], Christophe Giraud-Carrier[2], Pavel Brazdil[3], Carlos Soares[3]

[1] University of Houston, USA; vilalta@cs.uh.edu
[2] Brigham Young University, USA; cgc@cs.byu.edu
[3] LIACC/FEP, University of Porto; {pbrazdil, csoares}@liacc.up.pt.

Abstract:

Current data mining tools are characterized by a plethora of algorithms but a lack of guidelines to select the right method according to the nature of the problem under analysis. Producing such guidelines is a primary goal by the field of meta-learning; the research objective is to understand the interaction between the mechanism of learning and the concrete contexts in which that mechanism is applicable. The field of meta-learning has seen continuous growth in the past years with interesting new developments in the construction of practical model-selection assistants, task-adaptive learners, and a solid conceptual framework. In this paper, we give an overview of different techniques necessary to build meta-learning systems. We begin by describing an idealized meta-learning architecture comprising a variety of relevant component techniques. We then look at how each technique has been studied and implemented by previous research. In addition, we show how meta-learning has already been identified as an important component in real-world applications.

1 Introduction

Meta-learning 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 learning task (e.g., credit rating, medical diagnosis, mine-rock discrimination, fraud detection, etc.), learning at the meta-level is concerned with accumulating experience on the performance of multiple applications of a learning system. 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. Briefly stated, the field of meta-learning is focused on the relation between tasks or domains and learning strategies. In that sense, by learning or explaining what causes a learning system to be successful or not on a particular task or domain, we go beyond the goal of producing more accurate learners to the additional goal of understanding the conditions (e.g., types of example distributions) under which a learning strategy is most appropriate.

From a practical stance, meta-learning helps solve important problems in the application of machine learning (ML) and data mining (DM) tools, particularly in the area of classification and regression. First, the successful use of these tools outside the boundaries of research (e.g., industry, commerce, government) is conditioned on the appropriate selection of a suitable predictive model (or combination of models) according to the domain of application. Without some kind of assistance, model

selection and combination can turn into solid obstacles to end-users who wish to access the technology more directly and cost-effectively. End-users often lack not only the expertise necessary to select a suitable model, but also the availability of many models to proceed on a trial-and-error basis (e.g., by measuring accuracy via some re-sampling technique such as n-fold cross-validation). A solution to this problem is attainable through the construction of meta-learning systems. These systems can provide automatic and systematic user guidance by mapping a particular task to a suitable model (or combination of models).

Second, a problem commonly observed in the practical use of ML and DM tools is how to profit from the repetitive use of a predictive model over similar tasks. The successful application of models in real-world scenarios requires continuous adaptation to new needs. Rather than starting afresh on new tasks, one would expect the learning mechanism itself to re-learn, taking into account previous experience ([16],[41],[47],[52]). Again, meta-learning systems can help control the process of exploiting cumulative expertise by searching for patterns across tasks.

Our goal in this paper is to give an overview of different techniques necessary to build meta-learning systems. To provide some structure, we begin by describing an idealized meta-learning architecture comprising a variety of component techniques. We then show what role these techniques played in previous research. We hope that by proceeding in this way the reader can not only learn from past work, but in addition gain some insights concerning how to construct new meta-learning systems.

We also hope to show how recent advances in meta-learning are increasingly filling the gaps in the construction of practical model-selection assistants and task-adaptive learners, as well as in the development of a solid conceptual framework ([6],[7],[28]).

The paper is organized as follows. In the next section we illustrate an idealized meta-learning architecture and detail its constituent parts. In Section 3 we describe previous research in meta-learning and its relation to our architecture. Section 4 describes meta-learning tools that have been instrumental in real applications. Finally, Section 5 concludes the paper.

2 A Meta-Learning Architecture

In this section, we provide a general view of a software architecture that will be used as a reference to describe many of the principles and current techniques in meta-learning. Though not every technique in meta-learning fits into this architecture, such a general view helps us understand the challenges that need to be overcome before we can turn the techniques into a set of useful and practical tools.

Conceptually, our proposed meta-learning system can be divided into two modes of operation: acquisition and advisory, as detailed in the following sections.

2.1 Meta-Learning: Knowledge Acquisition Mode

During the knowledge acquisition mode, the main goal is to learn about the learning process itself. Figure 1 illustrates this mode of operation. We assume that the input to the system consists of datasets of examples (e.g., sets of pairs of feature vectors and classes; Fig. 1-A). Upon arrival of each dataset, the meta-learning system

invokes a component responsible for extracting dataset characteristics or meta-features (Fig. 1-B). The goal of this component is to gather information that transcends a particular domain of application. We look for information that can be used to generalize to other example distributions. Section 3.1 details current research pointing in this direction.

During the knowledge acquisition mode, the learning techniques (Fig. 1-C) do not exploit knowledge of previous results. Statistics derived from different learning strategies (e.g., a classifier or combination of classifiers, Fig. 1-D) and their performance (Fig. 1-E) may be used as a form of characterizing the task under analysis (Sections 3.1 and 3.2).

Information derived from the meta-feature generator and the performance evaluation module can be combined into a meta-knowledge base (Fig. 1-F). This knowledge base is the main result of the knowledge acquisition phase; it reflects experience accumulated across different tasks. Meta-learning is tightly linked to the process of acquiring and exploiting meta-knowledge. One can even say that advances in the field of meta-learning hinge on one specific question: how can we acquire and exploit knowledge about learning systems (i.e., meta-knowledge) to understand and improve their performance? As we describe current research in meta-learning we will be pointing to different forms of meta-knowledge.

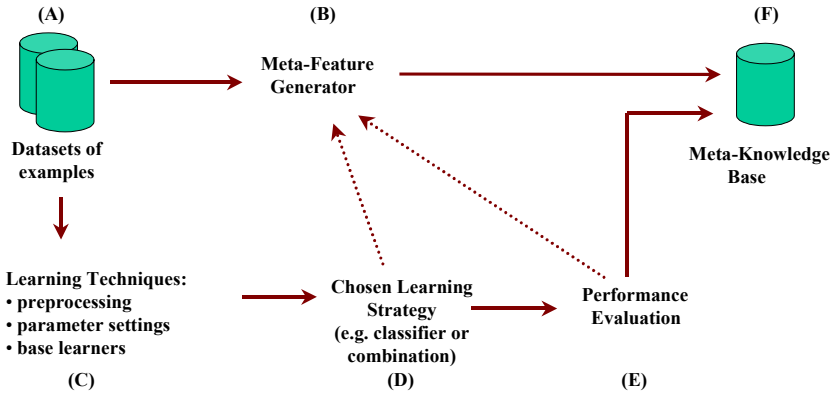

Figure 1. Meta-Learning: The Knowledge Acquisition Mode.

2.2 Meta-Learning: Advisory Mode

In the advisory mode, meta-knowledge acquired in the exploratory mode is used to configure the learning system in a manner that exploits the characteristics of the new data. Meta-features extracted from the dataset (Fig. 2-B) are "matched" with the meta-knowledge base (Fig. 2-F) to produce a recommendation regarding the best available learning strategy. At this point we move away from experimentation with the base learners to the ability to do informed model selection or combination of base learners (Fig. 2-C).

The effectiveness of the meta-learner increases as it accumulates meta-knowledge. The lack of experience at the beginning of the learner's existence compels the meta-learner to use one or more learning strategies without a clear preference for any one of them; experimenting with many different strategies is time consuming. However, as more training sets have been examined, we expect the expertise of the meta-learner to dominate the process of deciding which learning strategy suits best the characteristics of the current problem.

The nature of the match between the set of meta-features and the meta-knowledge base can have several interpretations. The traditional view poses this problem as a learning problem itself where a meta-learner is invoked to output an approximating function mapping meta-features to learning strategies (e.g., learning model). However, it is conceivable that the meta-learner could be subject to improvement through meta-learning ([43],[51]). Here, the matching process is not intended to modify our set of available learning techniques, but simply enables to select one or more strategies that seem effective given the characteristics of the dataset under analysis.

The final classifier (or combination of classifiers; Fig. 2-D) is selected based not only on its estimate of the generalization performance over the current dataset, but also on information derived from exploiting past experience. In this case, the system has moved from experimenting with different learning strategies (or choosing on at random) to the ability of selecting one dynamically.

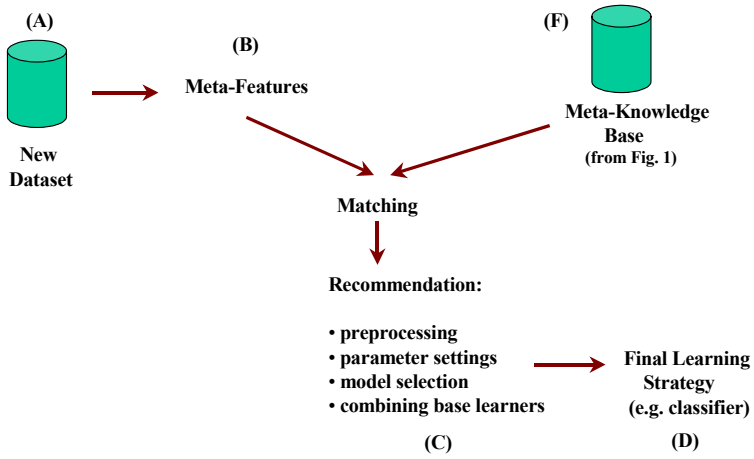

Figure 2. Meta-Learning: The Advisory Mode.

We will show how the constituent components conforming our two-mode meta-learning architecture can represent a variety of different techniques.

3 Techniques in Meta-Learning

In this section we describe previous research in meta-learning and in particular address the following specific research issues:

1. The characterization of datasets can be performed using a variety of statistical, information-theoretic, and model-based approaches (Section 3.1). Matching meta-features to predictive model(s) can help in model selection or ranking.
2. Information collected from the performance of a set of learning algorithms at the base level can be combined through a meta-learner (Section 3.2).
3. Within the learning-to-learn paradigm, a continuous learner can extract knowledge across domains or tasks to accelerate the rate of learning convergence (Section 3.3).
4. The learning strategy can be modified in an attempt to shift this strategy dynamically (Section 3.4). A meta-learner in effect explores not only the space of hypotheses within a fixed family set, but in addition the space of families of hypotheses.

3.1 Meta-Learning for Machine Learning

3.1.1 Dataset Characterization

A critical component of any meta-learning system needs to extract relevant information about the task under analysis (Fig. 1-B and 2-B). The central idea is that high-quality dataset characteristics or meta-features provide some information to differentiate the performance of a set of given learning strategies. We describe a representative set of techniques in this area.

Statistical and Information-Theoretic Characterization
Much work in dataset characterization has concentrated on extracting statistical and information-theoretic parameters estimated from the training set ([2],[21],[25],[31],[34],[46]). Measures include number of classes, number of features, ratio of examples to features, degree of correlation between features and target concept, average class entropy and class-conditional entropy, skewness, kurtosis, signal–to-noise ratio, etc. This work has produced a number of research projects with positive and tangible results (e.g., ESPRIT Statlog and METAL).

Model-Based Characterization
In addition to statistical measures, a different form of dataset characterization exploits properties of the induced hypothesis as a form of representing the dataset itself. As an example, one can build a decision tree from a dataset and collect properties of the tree (e.g., nodes per feature, maximum tree depth, shape, tree imbalance, etc.), as a means to characterize the dataset ([9],[38]).

Landmarking
Another source of characterization falls within the concept of landmarking ([8],[39]). The idea is to exploit information obtained from the performance of a set of simple

learners (i.e., learning systems with low capacity) that exhibit significant differences in their learning mechanism. The accuracy (or error rate) of these landmarkers is used to characterize a dataset and identify areas where each of the simple learners can be regarded as an expert.

Another idea related to landmarking is to exploit information obtained on simplified versions of the data (e.g. samples). Accuracy results on these samples serve to characterise individual datasets and are referred to as *sub-sampling landmarks*. This information is subsequently used to select an appropriate learning algorithm ([24],[45]).

3.1.2 Mapping Datasets to Predictive Models

An important and practical use of meta-learning is the construction of a mechanism that maps an input space composed of datasets or applications to an output model space composed of predictive models. Criteria such as accuracy, storage space, and running time can be used for performance assessment ([27]; Fig. 1-E). Several approaches have been developed in this area.

Hand-Crafting Meta Rules

Using human expertise and empirical evidence, a number of meta-rules matching domain characteristics with learning techniques may be crafted manually [14]. 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. Crafting rules manually has the disadvantage of failing to identify many important rules. As a result most research has focused on learning these meta-rules automatically as explained next.

Learning at the Meta-Level

The characterization of a dataset is a form of meta-knowledge (Fig. 1-F) that is commonly embedded in a meta-dataset as follows. After learning from several tasks, one can construct a meta-dataset where each element pair is made up of the characterization of a dataset (meta-feature vector) and a class label corresponding to the model with best performance on that dataset. A learning algorithm can then be applied to this well-defined learning task to induce a hypothesis mapping datasets to predictive models.

A variation to the approach above is to look at the neighbourhood of a query in the space of meta-features. When a new query dataset is presented, the k-nearest neighbour instances (i.e., datasets) around this dataset are identified to select the model with best average performance [11].

Instead of mapping a task or dataset to a predictive model, a different approach consists of selecting a model for each individual query example. The idea is similar to the nearest-neighbour approach: select the model displaying best performance around the neighbourhood of the query example [32].

Ranking Models

Rather than mapping a dataset to a single predictive model, one may also produce a ranking over a set of different models. One can argue that such rankings are more flexible and useful to users. In a practical scenario, the advice should not be limited to a single item; this could lead to problems if the suggested final model happens to

be unsatisfactory. Rankings provide alternative solutions to users who may wish to incorporate their own expertise or any other criterion (e.g., financial constraints) into their decision-making process. Various approaches have been suggested attacking the problem of ranking predictive models ([11],[12],[25],[36]).

3.2 Combining Base-Learners

Another approach to meta-learning consists of learning from base learners. The idea is to make explicit use of information collected from the performance of a set of learning algorithms at the base level; such information is then incorporated into the meta-learning process.

Stacked Generalization

Meta-knowledge (Fig. 1-F) can incorporate predictions of base learners, a process known as stacked generalization [54]. The process works under a layered architecture. Each of a set of base-classifiers is trained on a dataset; the original feature representation is then extended to include the predictions of these classifiers. Successive layers receive as input the predictions of the immediately preceding layer and the output is passed on to the next layer. A single classifier at the topmost level produces the final prediction. Most research in this area focuses on a two-layer architecture ([13],[18],[44] etc.).

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). Research in this area investigates what base-learners and meta-learners produce best empirical results (e.g., [20],[26]); how to represent class predictions (class labels versus class-posterior probabilities [48]); and how to define meta-features ([3],[15]).

Boosting

A popular approach to combining base learners is called boosting ([22],[23],[30]). The basic idea is to generate a set of base learners by generating variants of the training set. Each variant is generated by sampling with replacement under a weighted distribution. This distribution is modified for every new variant by giving more attention to those examples incorrectly classified by the most recent hypothesis.

Boosting is considered a form of meta-learning because it takes into consideration the predictions of each hypothesis over the original training set so as to progressively improve the classification of those examples where the last hypothesis failed.

Meta-Decision Trees

Another approach in the field of learning from base learners consists of combining several inductive models by means of induction of meta-decision trees [49]. The general idea is to build a decision tree where each internal node is a meta-feature and each leaf node corresponds to a predictive model. Given a new example, a meta-decision tree indicates the model that appears most suitable in predicting its class label.

Composition of Inductive Applications
The CAMLET system composes models using components with different biases [1]. CAMLET is based on a template that abstracts the process of inductive learning. For a given data set this template is instantiated using components that are organized according to different repositories. The final model is obtained through an iterative search for the best components attached to this template.

Meta-learning for Pre-processing
Another application of meta-learning is done before a learning algorithm is applied, as a form of data pre-processing. Leite & Brazdil [55] propose a meta-learning approach to reduce the number of samples during progressive sampling. The process stops when the learning curve has levelled off. The corresponding sample is referred to as the *stopping point*. The aim of the method is to predict the stopping point in the dataset under study. The method compares the first few points on the learning curve constructed for a given learning algorithm and the dataset under study. The datasets with most similar curves are selected and the corresponding stopping points are used to estimate the stopping point for the current dataset. This information can be used to skip investigation of some of the samples and hence leads to time savings.

3.3 Inductive Transfer and Learning to Learn

We have mentioned before that learning should not be viewed as an isolated task that starts from scratch on every new problem. As experience accumulates, the learning mechanism is expected to perform increasingly better. One approach to simulate the accumulation of experience is by transferring meta-knowledge across domains or tasks. This process is known as *inductive transfer* [41]. The goal here is not to match meta-features with a meta-knowledge base (Fig. 2), but simply to incorporate the meta-knowledge into the new learning task.

A review of how neural networks can learn from related tasks is provided by Pratt & Jennings [42]. Caruana [16] shows the reasons explaining why learning works well in the context of neural networks using backpropagation. In essence, 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 [47] proposes a learning algorithm that groups similar tasks into clusters. A new task is assigned to the most related cluster; inductive transfer takes place when generalization exploits information about the selected cluster.

3.3.1 A Theoretical Framework of Learning-to-Learn

Several studies have provided a theoretical analysis of the learning-to-learn paradigm within a Bayesian view [6], and within a Probably Approximately Correct (PAC) view [7]. In the PAC view, meta-learning takes place because the learner is not only looking for the right hypothesis in a hypothesis space, but in addition is searching for the right hypothesis space in a family of hypothesis spaces. Both the VC dimension and the size of the family of hypothesis spaces can be used to derive bounds on the number of tasks, and the number of examples on each task, required to ensure with high probability that we will find a solution having low error on new training tasks.

3.4 Dynamic Bias Selection

A field related to the idea of learning-to-learn is that of dynamic bias selection. This can be understood as the search for the right hypothesis space or concept representation as the learning system encounters new tasks. The idea, however, departs slightly from our architecture; meta-learning is not divided into two modes (i.e., knowledge-acquisition and advisory), but rather occurs on a single step. In essence, the performance of a base learner (Fig. 1-E) can trigger the need to explore additional hypothesis spaces, normally through small variations of the current hypothesis space.

As an example, DesJardins & Gordon [19] develop a framework for the study of dynamic bias as a search in different tiers. Whereas the first tier refers to a search over a hypothesis space, additional tiers search over families of hypothesis spaces. Other approaches to dynamic bias selection are based on changing the representation of the feature space by adding or removing features ([29],[50]). Alternatively, Baltes [5] 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.

A slightly different approach is to look at dynamic-bias selection as a form of data variation, but as a time-dependent feature [53]. The idea is to perform online detection of concept drift with a single base-level classifier. The meta-learning task consists of identifying contextual clues, which are used to make the base-level classifier more selective with respect to training instances for prediction. Features that are characteristic of a specific context are identified and contextual features are used to focus on relevant examples (i.e., only those instances that match the context of the incoming training example are used as a basis for prediction).

4 Meta-Learning for KDD and Data Mining: Tools and Applications

The process of knowledge discovery from databases (KDD) includes several steps (e.g., [17]), such as understanding the problem domain, selecting data sources, data cleaning and pre-processing, data reduction and projection, task selection, algorithm or model selection, model evaluation and deployment. Until now our focus has been on the use of meta-learning for model selection. However, meta-learning can be instrumental to other steps as well.

Here we describe some tools having industrial applications where meta-learning has served to provide useful recommendations. In addition, we describe two other approaches that support the development of solutions for data mining applications that can benefit from the use of meta-learning.

4.1 METAL DM Assistant

The METAL Data Mining Assistant (DMA) is the result of a large European Research and Development project broadly aimed at the development of methods and tools for providing support to users of machine learning and data mining technology [33]. DMA is a web-enabled prototype assistant system that supports users for model

selection. The project had as its main goal improving the use of data mining tools and in particular to provide savings in experimentation time.

The k-NN Ranking Method

DMA provides recommendations in the form of rankings (Section 3.1). Instead of delivering a single model candidate, it produces an ordered list of models, sorted from best to worst, according to a weighted combination of parameters such as accuracy and training time.

Given a new dataset, DMA computes a set of statistical and information-theoretic measures (Section 3.1). Those measures define a space from which the most similar datasets in the Meta-Knowledge Base (Fig. 1-F) are retrieved using a k-NN method. For each of the selected datasets, a ranking of the candidate models is generated based on performance criteria (accuracy and learning time). The rankings obtained are aggregated to generate the final recommended ranking. DMA incorporates more than one ranking method. One method exploits a ratio of accuracies and times [12]. Another, referred to as DCRanker [11], is based on a technique known as Data Envelopment Analysis ([4],[37] etc.).

The user determines the relative importance of the accuracy and time which is most appropriate for the current application. An example of a ranking of 10 well-known algorithms, which is recommended for the *letter* dataset is presented in Table 1. The table is quite similar to the information provided on-line by DMA. Column 2 shows the recommended ranking[1]. The information shown in the other columns is discussed further on.

Table 1. Example of ranking of 10 algorithms recommended for the *letter* dataset and the corresponding true ranking.

Algorithm	Recommended Rank	Target Rank	Accuracy %	Time s
Boosted C5.0	1	1	95.3	77
IB1	2	2	93.6	163
Linear Discriminant	3	8	70.2	2
Ltree	4	4	86.9	397
C5.0 (rules)	5	3	88.8	222
C5.0 (tree)	6	5	87.9	8
Naive Bayes	6	9	64.4	10
RIPPER	8	6	86.2	1249
Radial-Basis Function Network	9	10	43.9	4946
MultiLayer Perceptron	10	7	79.8	3998

Evaluation of Rankings of Algorithms

[1] The results depend on the options used. Here more importance was given to accuracy than to time. The predicted ranking was generated on the basis of 3 similar datasets. All algorithms were used with default settings.

Different approaches exist to evaluate methods that predict rankings of learning algorithms [12]. One consists of calculating the resulting ranking accuracy. The accuracy is given by the similarity between the recommended ranking (column 2 of Table 1) and the target ranking on the corresponding dataset (column 3). The target ranking is based on estimates of the performance of the algorithms on the dataset (e.g., by cross-validation). The corresponding values of accuracy and time (the sum of train and run time) are shown in subsequent columns.

The similarity between the rankings can be measured using the common *rank correlation coefficients* (e.g., Spearman's), or weighted coefficients that assign more importance to higher ranks [40].

An additional approach is based on an assumption that the top *N* algorithms in the ranking have been examined by the user (i.e. we assume that he carried out both training and testing). The objective is to identify the algorithm with the best performance (the highest accuracy). For the sake of argument, let us assume that N is, say, 3. We are interested to estimate also the corresponding computational effort. We can thus plot a point in a graph, showing the best accuracy achieved on one of the axes (say, Y) and the number of algorithms executed (or the corresponding time of using the 3 algorithms) on the other axis (i.e. X). This process can be repeated for different values of N. Different ranking approaches can thus be compared by looking at the resulting curves in this kind of graph [12].

Applications of the DMA

DMA is providing a practical and effective tool to users in need for assistance in model selection. In addition, the results obtained from a large number of controlled experiments on both synthetic and real-world datasets are readily available. Besides, DMA has been instrumental as a decision support tool within DaimlerChrysler and in the field of Automotive Industry [11].

As a publicly available tool, DMA's success has surpassed the initial expectations, with a few hundred registered users and dozens of datasets that were uploaded since it was turned public.

4.2 Ranking Processes with IDEA

The Intelligent Discovery Electronic Assistant (IDEA) is an automated assistant for the KDD process elaborated by Bernstein & Provost [10]. The goal is to support several steps in the KDD process, from data cleaning and pre-processing to deployment. IDEA consists of two components. First, a *plan generator* uses an ontology to build a list of processes that are appropriate for a specific task. Here, a process is a chain of operations (e.g., a pre-processing method followed by a learning algorithm and a post-processing method). Next, a *heuristic ranker* orders the processes using heuristics. The heuristic rankings are knowledge-based and can take into account user's preferences (e.g., speed vs. accuracy). In the current implementation of IDEA, rankings are fixed. However, IDEA is independent of the ranking method and, thus it could possibly be improved by incorporating meta-learning to generate rankings based on past performance.

4.3 Support of Pre-Processing with MiningMart

Another interesting problem has been addressed through the MiningMart project [35]. The goal is to reuse successful pre-processing steps, organized in the form of processes organized in the form of a partially ordered graph. The meta-data describing the data and the pre-processing steps used in different applications are organized into ontologies. The user searches through the meta-data base for the processes that seem most appropriate for the problem at hand. Next, the user describes the mapping between the user's new problem and the previous ones, including the processes retrieved. The system then generates pre-processing steps for the new problem that can be executed automatically. Again, meta-learning could be used to help the user match the current problem with the most suitable one in the meta-data base.

5 Conclusions

In this paper, we have discussed a generic architecture of a meta-learning system and showed how different components interact. We have provided a survey of relevant research in the field, together with a description of available tools and applications.

One important research direction in meta-learning consists of searching for alternative meta-features in the characterization of datasets (Section 3.1). A proper characterization of datasets can elucidate the interaction between the learning mechanism and the task under analysis. Current work has only started to unveil relevant meta-features; clearly much work lies ahead. For example, many statistical and information-theoretic measures adopt a global view of the dataset under analysis; meta-features are obtained by averaging results over the entire training set, implicitly smoothing the actual distribution (e.g., class-conditional entropy is estimated by projecting all training examples over a single feature dimension.). There is a need for alternative and more detailed descriptors of the example distribution in a form that highlights the relationship to the learner's performance.

Using data samples in conjunction with a principled method of carrying out tests seems another promising direction that should be explored in future.

We conclude this paper by emphasizing the important role of meta-learning as an assistant tool in the tasks of model selection and combination (Section 4). Classification and regression tasks are common in daily business practice across a number of sectors. Hence, a decision support offered by a meta-learning assistant has the potential of bearing a strong impact in future applications. In particular, since prior expert knowledge is often expensive and not always readily available, and besides, subject to bias and personal preferences, meta-learning can serve as a useful complement through the automatic accumulation and exploitation of meta-knowledge.

Acknowledgements

The authors of the University of Porto wish to acknowledge the support under Portuguese Pluriannual Programme and Program POSI.

References

1. Abe H., Yamaguchi T. Constructing Inductive Applications by Meta-Learning with Method Repositories. *Progress in Discovery Science, Final Report of the Japanese Discovery Science Project*, 576-585. Springer-Verlag, 2002.
2. Aha D. W. Generalizing from Case Studies: A Case Study. In *Proceedings of the Ninth International Workshop on Machine Learning*, 1-10, Morgan Kaufman, 1992.
3. Ali K., Pazzani M. J. Error Reduction Through Learning Model Descriptions. *Machine Learning, 24:*173:202, 1996.
4. Andersen, P., Petersen, N.C. A Procedure for Ranking Efficient Units in Data Envelopment Analysis. *Management Science, 39(10):*1261-1264, 1993.
5. Baltes J. Case-Based Meta Learning: Sustained Learning Supported by a Dynamically Biased Version Space. In *Proceedings of the Machine Learning Workshop on Biases in Inductive Learning,* 1992.
6. Baxter, J. Theoretical Models of Learning to Learn. In *Learning to Learn, Chapter 4*, 71-94, MA: Kluwer Academic Publishers, 1998.
7. Baxter, J. A Model of Inductive Learning Bias. *Journal of Artificial Intelligence Research*, *12*:149-198, 2000.
8. Bensusan, H., Giraud-Carrier, C. Discovering Task Neighbourhoods Through Landmark Learning Performances. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 325,330, 2000.
9. Bensusan H., Giraud-Carrier C., Kennedy C. J. A Higher-Order Approach to Meta-Learning. In *Proceedings of the ECML-2000 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination,* 109-118. 2000.
10. Bernstein A., Provost F. An Intelligent Assistant for the Knowledge Discovery Process. In *Proceedings of the IJCAI-01 Workshop on Wrappers for Performance Enhancement in KDD,* 2001.
11. Berrer, H., Paterson, I., Keller, J. Evaluation of Machine-learning Algorithm Ranking Advisors. In *Proceedings of the PKDD-2000 Workshop on Data-Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions,* 2000.
12. Brazdil, P., Soares, C., Pinto da Costa, J. Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. *Machine Learning, 50(3):*251-277, 2003.
13. Breiman, L. Stacked Regressions. Machine Learning, 24:49-64, 1996.
14. Brodley, C. Recursive Automatic Bias Selection for Classifier Construction. *Machine Learning, 20,* 1994.
15. Brodley C., Lane T. Creating and Exploiting Coverage and Diversity. In *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models,* 8-14, 1996.
16. Caruana, R. Multitask Learning. Second Special Issue on Inductive Transfer. *Machine Learning, 28:*41-75, 1997.
17. Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., Wirth R. CRISP-DM 1.0: Step-by-Step Data Mining Guide, SPSS, 2000.
18. Chan P., Stolfo S. On the Accuracy of Meta-Learning for Scalable Data Mining. *Journal of Intelligent Integration of Information,* Ed. L. Kerschberg, 1998.
19. DesJardins M., Gordon D. F. Evaluation and Selection of Biases in *Machine Learning, 20:*5-22, 1995.
20. Dzeroski S., Zenko B. Is Combining Classifiers Better than Selecting the Best One? *Machine Learning, 54:*255-273, 2004.

21. Engels, R., Theusinger, C. Using a Data Metric for Offering Preprocessing Advice in Data-mining Applications. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence,* 1998.
22. Freund, Y., Schapire, R. E. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning,* 148-156, 1996.
23. Friedman, J., Hastie, T., Tibshirani, R. Additive Logistic Regression: A Statistical View of Boosting. *Annals of Statistics, 28:*337-387, 2000.
24. Fürnkranz, J., Petrak J. An Evaluation of Landmarking Variants, in C. Giraud-Carrier, N. Lavrac, Steve Moyle, and B. Kavsek, editors, *Working Notes of the ECML/PKDD 2000 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning,* 2001.
25. Gama, J., Brazdil, P. A Characterization of Classification Algorithms. In *Proceedings of the Seventh Portuguese Conference on Artificial Intelligence,* 189-200, 1995.
26. Gama, J., Brazdil P. Cascade Generalization, *Machine Learning, 41(3),* Kluwer, 2000.
27. Giraud-Carrier, C. Beyond Predictive Accuracy: What? In *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation,* 78-85, 1998.
28. Giraud-Carrier, C., Vilalta, R., Brazdil, P. Introduction to the Special Issue on Meta-Learning. *Machine Learning, 54:*187-193, 2004.
29. Gordon D. Perlis D. Explicitly Biased Generalization. *Computational Intelligence, 5:*67-81, 1989.
30. Hastie, T., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series, 2001.
31. Hilario, M., Kalousis, A. Fusion of Meta-Knowledge and Meta-Data for Case-Based Model Selection. *In Proceedings of the Fifth European Conference on Principles and Practice of Knowledge Discovery in Databases,* 2001.
32. Merz C. Dynamical Selection of Learning Algorithms. *Learning from Data: Artificial Intelligence and Statistics,* D. Fisher and H. J. Lenz (Eds.), Springer-Verlag, 1995.
33. Metal. A Meta-Learning Assistant for Providing User Support in Machine Learning and Data Mining http://www.metal-kdd.org/ 2002.
34. Michie, D., Spiegelhalter, D. J., Taylor, C.C. *Machine Learning, Neural and Statistical Classification.* England: Ellis Horwood, 1994.
35. Morik K., Scholz M. The MiningMart Approach to Knowledge Discovery in Databases. Intelligent Technologies for Information Analysis, 2003.
36. Nakhaeizadeh, G., Schnabel, A. Development of Multi-criteria Metrics for Evaluation of Data-mining Algorithms. In *Proceedings of the Third International Conference on Knowledge Discovery and Data-Mining,* 1997.
37. Paterson, I. New Models for Data Envelopment Analysis, Measuring Efficiency with the VRS Frontier. Economics Series No. 84, Institute for Advanced Studies, Vienna, 2000.
38. Peng, Y., Flach, P., Brazdil, P., Soares, C. Decision Tree-Based Characterization for Meta-Learning. In *Proceedings of the ECML/PKDD'02 Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning,* 111-122, 2002.
39. Pfahringer, B., Bensusan, H., Giraud-Carrier, C. Meta-learning by Landmarking Various Learning Algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning,* 743-750, 2000.
40. Pinto da Costa J., Soares C. A Weighted Rank Measure of Correlation, *Australian and New Zealand Journal of Statistics,* to be published, 2004.
41. Pratt, L., Thrun, S. Second Special Issue on Inductive Transfer. *Machine Learning, 28,* 1997.

42. Pratt S., Jennings B. A Survey of Connectionist Network Reuse through Transfer. In *Learning to Learn, Chapter 2,* 19-43, Kluwer Academic Publishers, MA, 1998.
43. Schmidhuber J. Discovering Solutions with Low Kolmogorov Complexity and High Generalization Capability. In *Proceedings of the Twelve International Conference on Machine Learning,* 488-49, Morgan Kaufman, 1995.
44. Skalak, D. *Prototype Selection for Composite Nearest Neighbor Classifiers.* PhD thesis, University of Massachusetts, Amherst, 1997.
45. Soares, C., Petrak, J., Brazdil, P. Sampling-Based Relative Landmarks: Systematically Test-Driving Algorithms before Choosing. In *Proceedings of the Tenth Portuguese Conference on Artificial Intelligence,* Springer, 2001.
46. Sohn, S.Y. Meta Analysis of Classification Algorithms for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 21(11):1137-1144, 1999.
47. Thrun, S. Lifelong Learning Algorithms. In *Learning to Learn,* Chapter 8, 181-209, MA: Kluwer Academic Publishers, 1998.
48. Ting, K. M., Witten I. H. Stacked generalization: When does it work?. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence,* 866-873, 1997.
49. Todorovski, L., Dzeroski, S. Combining Classifiers with Meta Decision Trees. *Machine Learning, 50 (3):*223-250, 2003.
50. Utgoff P. 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, 1986.
51. Vilalta, R. Research Directions in Meta-Learning: Building Self-Adaptive Learners. In *Proceedings of the International Conference on Artificial Intelligence*, 2001.
52. Vilalta, R., Drissi, Y. A Perspective View and Survey of Meta-Learning. *Journal of Artificial Intelligence Review, 18(2)*:77-95, 2002.
53. Widmer, G. Tracking Context Changes through Meta-Learning. *Machine Learning, 27(3):*259-286, 1997.
54. Wolpert D. Stacked Generalization. *Neural Networks*, **5**:241-259, 1992.
55. Leite R. and Brazdil P. Improving Progressive Sampling via Meta-learning on Learning Curves. *Machine Learning – ECML-2004*. Springer Verlag, 2004.