# The Data Mining Advisor: Meta-learning at the Service of Practitioners

Christophe Giraud-Carrier
Department of Computer Science, Brigham Young University, Provo, UT 84602

## Abstract

*In order to make Machine Learning algorithms more usable, our community must be able to design robust systems that offer support to practitioners. In the context of classification, this amounts to developing assistants, which deal with the increasing number of models and techniques, and give advice dynamically on such issues as model selection and method combination. This paper briefly reviews the potential of meta-learning in this context and reports on the early success of a Web-based Data Mining assistant.*

## 1 Introduction

One of the results of the NFL Theorems [26, 34, 33] for classification tasks in Machine Learning is that no algorithm is better overall than any other. Although rather theoretical, since it applies to the universe of all tasks, this result has two important practical implications for both algorithm designers and practitioners. When a designer introduces a novel classification algorithm, how does she position it in the exisiting algorithm landscape? When a practitioner is faced with a new classification task for which she seeks a model with high accuracy, how does she know which algorithm to use? In either case, the only two viable alternatives in light of the NFL Theorems are:

1. *Closed Classification World Assumption (CCWA)*: She assumes that all classification tasks likely to occur in real applications form some well-defined subset of the universe. As a designer, she shows that her novel algorithm performs better than others on that set. As a practitioner, she picks any of the algorithms that performs well on that set.

2. *Open Classification World Assumption (OCWA)*: She assumes no structure on the set of classification tasks likely to occur in real applications. As a designer, she characterizes as precisely as possible the class of tasks on which her novel algorithm outperforms others. As a practitioner, she has some way of determining which algorithm(s) will perform well on her specific task(s).

Interestingly, the widely-used approach consisting in benchmarking algorithm against well-known repositories (e.g., UCI [7], UCR [18]) tends to implicitly favor the CCWA. Yet, this remains at best a conjecture. In a simple experiment, we applied 20 algorithms from Weka [31], representing all known model classes, to 52 UCI classification datasets. We found that 1) no algorithm significantly outperforms all others on all tasks, and 2) there are tasks on which all algorithms perform rather poorly (i.e., only marginally better than chance). To date, no one has offered a characterization of "real-life" classification tasks. And evidence suggests that this is by no means a trivial task. To compound the problem for practitioners, most efforts in algorithm design seem to share the same oblivious pattern:

1. They propose new algorithms that overcome known limitations. Yet, unless one accepts the CCWA, this simply shifts the original question of how to overcome the targeted limitations to the equally difficult question of determining what applications the proposed approach works well on.

2. They "promote" new algorithms on the basis of limited empirical results, leaving the burden of proof to the users. It is not trivial to know how well any new approach will generalize beyond the problems it has been tested against so far.

A recent example provides an eloquent illustration. In [19], the authors argue that too many mining algorithms incorporate parameters that are difficult to set, and introduce degrees of freedom that may cause spurious effects as well as hinder the repeatability of experiments. They then proceed to show how Kolmogorov complexity can be successfully used to design compression-based, parameter-free algorithms. Through experiments, they demonstrate that "The accuracy of our approach *can be* greatly superior to those of parameter-laden algorithms, even if we allow these algorithms to search exhaustively over their parameter spaces" (emphasis added). Since, by the NFL Theorems, parameter-free approaches are no better overall than other approaches, laden or not with parameters, the real question is not whether such approaches *can be* superior to others, but *under what conditions* they may be expected so to be.

Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05)
0-7695-2495-8/05 $20.00 © 2005 IEEE

Thus, the authors' intent to "theoretically and empirically investigate the limitations on object sizes that we can meaningfully work with using our proposed approach."

Lacking expertise and/or support, users faced with a new classification data mining task often resort to a trial-and-error process to select the most suitable model. Clearly, trying all possible options is impractical, and choosing the option that "appears" most promising is likely to yield a sub-optimal solution. What is needed is an informed search process to reduce the amount of experimentation while avoiding the pitfalls of local optima. Informed search requires meta-knowledge about the precise conditions under which a given method is better than others for a given task. In turn, such meta-knowledge may be usefully incorporated into data mining assistants that offer support to non-expert practitioners. In this paper, we revisit the use of meta-learning as a robust mechanism to build meta-knowledge about model selection in classification. We then outline the technical specifications and functionality of the publicly available Data Mining Advisor [10], reporting key usage figures since its inception. Finally, we conclude with several directions for future research and development.

## 2 Meta-learning for Model Selection

Meta-learning, in the context of model selection, consists of applying learning mechanisms to the problem of mapping classification tasks to algorithms. Let $L$ be a set of learning algorithms for classification and $T$ be a set of classification tasks such that for each $t \in T$, $b_L(t)$ represents the algorithm in $L$ that performs best on $t$. Since classification tasks may be unwieldy to handle directly, we resort to some characterization of tasks and the meta-learner actually learns a mapping from characterizations to algorithms. For each classification task $t$, we denote by $c(t)$ the characterization of $t$ by some fixed mechanism. Then, meta-learning takes the set $\{< c(t), b_L(t) >: t \in T\}$ as a training set and induces a meta-model that, for each new classification task, predicts the model from $L$ that will perform best. Note that unless $\forall S \neq S' : c(S) = c(S') \Rightarrow b(S) = b(S')$, then the meta-training set may be noisy and meta-learning may in turn be sub-optimal. We ignore this issue here.

Alternatively, one may build a meta-model that predicts a ranking of algorithms from $L$ (e.g., [6, 9]). This approach reduces the brittleness of the meta-model. Assume the model predicted best for some new classification task $t'$ results in what appears to be a poor performance on $t'$. In the single-model prediction approach, the user has no further information as to what other model to try. In the ranking approach, the user may try the second best, third best, and so on, in an attempt to improve performance. Furthermore, ranking makes it easier to include qualitative criteria, such as comprehensibility, in the selection process.

We note at the onset that one of the consequences of the NFL Theorems for meta-learning is that, unless we accept that real-life classification tasks do not range over the entire universe of possible tasks or, similarly, that tasks are not uniformly distributed, there is no hope of finding a meta-model capable of usefully informing the choice of base-level methods. This is because, across all tasks, any meta-method of selection (e.g., cross-validation) will perform just as well as any other, including one that consistently selects the same base-level method, on average. This means that, even in the context of the OCWA, we must assume that only some areas of the base-level learning space are of interest, and the purpose of a meta-model is to accurately assign tasks to these areas.

One of the challenges of meta-learning is the construction of the training set, i.e, $< c(t), b_L(t) >$ pairs for some base level learning tasks. This raises issues with: 1) the choice of the characterization mechanism $c$, 2) the choice of the set of learners $L$, 3) the collection of representative tasks, and 4) the cost of computing $c(t)$ and $b_L(t)$ for each task. We briefly discuss each in turn in the following sections. Note that we ignore here the issue of choosing a meta-learner. We will assume that any learner whose expected generalization accuracy is better than random is acceptable.

### 2.1 Characterization Mechanism

As in any learning task, the characterization of the examples plays a crucial role in enabling learning. In particular, the features used must have some predictive power. So far, three main classes of characterization have been proposed.

- Statistics: A number of statistical and information-theoretic measures are extracted from the dataset, such as number of examples, skewness, mutual entropy, etc. (e.g., see [22, 11, 27]).
- Landmarking: The performances of simple learners, known as landmarkers, are computed on the dataset using cross-validation (e.g., see [3, 24, 12]). The idea is that landmarkers serve as signposts of the performance of the full-fledged target learners in $L$. Interestingly, this notion was first mentioned in [16], where the author, although not concerned by meta-learning, stated that "it may be possible to use the performance of 1-rules to predict the performance of the more complex hypotheses produced by standard learning systems."
- Model-based Properties: Models induced on the dataset are used as indicators of the underlying properties of the dataset. To date, only decision trees have been used [2, 4, 23].

Because of the NFL theorems, no characterization is better than any other on average. In fact, one of the consequences of the NFL theorems is that training data alone is

Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05)
0-7695-2495-8/05 $20.00 © 2005 IEEE

unable to inform the choice of a learner. However, the results of [32] suggest that the size of the training set and the size of the input space play a crucial role in determining the difference between algorithms. Since in practice, these are usually different, one may expect to capture sufficient information from these and other measures to discriminate among learners. No definitive conclusion has been reached as to which of the proposed mechanisms is best.

### 2.2 Choice of Base-level Learners

Although no learner is universal, each learner has its own area of expertise, defined as the set of learning tasks on which it performs well [3]. Since the role of the meta-model is to predict which algorithm is most likely to perform best on each new task, one should select base learners with complementary areas of expertise. The two issues in this choice are coverage and size. In principle, one should seek the smallest set of learners that is most likely to ensure a reasonable coverage of the base-level learning space. Minimizing the size ensures efficiency (see computational cost below), whilst maximizing coverage increases effectiveness.

In practice, it is non-trivial to ensure both coverage and a minimal set of learners. In fact, our experiments on the space of binary classification tasks of three Boolean variables (i.e., the complete space of 256 functions from $\{0,1\}^3$ to $\{0,1\}$) suggest that, from 26 applicable algorithms from Weka [31], a subset of 7 is sufficient to obtain maximal coverage, but 9 tasks still remain uncovered (i.e., none of the learners is better than chance on these). Short of designing effective learners for these tasks, one may hope that the performance gain obtained from the meta-model's correct predictions on covered tasks exceeds the loss incurred on uncovered tasks. Given that one is rarely interested in solving all possible learning tasks, one may further hope that the uncovered tasks are unlikely to arise in practice.

In order to approximate the best situation, we recommend that base learners have different biases by choosing representatives from varied model classes. This is consistent with the approach discussed in [28], where it is argued that experts generally first select a model class (e.g., decision trees or neural networks) on the basis of high-level characteristics of the task (e.g., noise level, attribute type, etc.), and then select the most promising algorithm(s) in that class. The more varied the biases, the greater the coverage.

### 2.3 Meta Training Set Construction

Meta-learning requires training meta-data, i.e., data about base level learning tasks. Unfortunately, the number of accessible, documented, real-world classification tasks is less than 100 today. Such a small sample poses a challenge for learning, which may be addressed either by augmenting the training set through systematic generation of synthetic base level tasks, or by taking the view that the model selection task is inherently incremental and treating it as such.

The first approach assumes that we know something about the kinds of tasks that occur in nature so that we are able to generate tasks with adequate characteristics. The second approach makes no such assumption but results in slower learning since classification tasks become available over time. On the other hand, it naturally adapts to reality, extending to new areas of the base level learning space only when tasks from these areas actually arise.

### 2.4 Computational Cost

The issue of computational cost is a consequence of the others and the price to pay to be able to perform model selection learning at the meta-level. We argue however that, in order to be justifiable, the cost of computing $c(t)$ should be significantly lower than the cost of computing $b_L(t)$. Otherwise, even if the meta-model is very accurate, it has little value as the user would be better off trying all algorithms and selecting the best one, which clearly defeats the purpose. The characterization mechanisms listed above all satisfy this condition.

### 2.5 Practical Implications

Although much remains to be done, results suggest the suitability of meta-learning for model selection (e.g., see [5, 8, 17, 15, 24]). From a practical standpoint, meta-learning contributes to the successful use of Data Mining tools outside the research arena, in industry, commerce, and government. Without some kind of assistance, model selection can turn into a serious road-block for 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). Meta-learning systems can provide automatic and systematic user guidance by mapping a particular task to a suitable model.

Most general-purpose commercial Data Mining packages (e.g., SAS' Enterprise Miner, SPSS' Clementine, IBM's Intelligent Miner, Megaputer's PolyAnalyst, Angoss' KnowledgeStudio, etc.) consist of collections of algorithms – often originating in the public domain and re-implemented with rather minor, if any, proprietary extensions – wrapped in a user-friendly graphical interface. While such tools facilitate access to algorithms, they generally offer no real decision support to non-expert end-users. Model selection assistants based on meta-learning could be

integrated naturally in future versions of these tools. In what follows, we briefly report on one such prototypical assistant.

## 3 The Data Mining Advisor

The Data Mining Advisor (DMA) is a prototype web-enabled assistant system, based on meta-learning, that gives users support with model selection (see http://www.metal-kdd.org). The DMA is one of the deliverables of European ESPRIT Framework IV's METAL Project (Nr. 26.357), a 3-year project broadly aimed at the development of methods and tools for providing support to users of machine learning and data mining technology [21]. One of the main goals of the DMA is to improve the utility of data mining tools and in particular to provide significant savings in experimentation time. We briefly outline the specifications and functionality of the DMA, and report on its usage since its inception. Both theoretical underpinnings and experiments that lead to current design decisions are found in the references.

### 3.1 Technical Specifications

In the context of the above formulation, the DMA works under the following constraints:

- $c(t)$ consists of statistical and information-theoretic measures, computed from the dataset, using the Data Characterization Tool (DCT) [20], version 3.2.
- $L$ consists of 10 widely-used classification algorithms from different model classes: C5.0 Trees, C5.0 Rules, C5.0 with boosting, Naive Bayes, IB1, MLP, RBF Network, Ripper, Linear Discriminant, and Ltree.
- The meta-learner is $k$-NN, with $k = 3$.
- The recommendation takes the form of a ranking of the 10 algorithms in $L$, obtained by aggregating the rankings of the query task's nearest neighbors.

The DMA's recommendation is based on both predictive accuracy and training time. The relationship between accuracy and time takes the simple form of a trade-off: "$AccD\%$ accuracy for a 10 times speed-up/slow-down." In the current implementation, three pre-defined settings are available, corresponding to $AccD = 0.1$ when the emphasis is on accuracy, $AccD = 10$ when the emphasis is on time, and $AccD = 1$ for a more aggressive compromise between accuracy and time, respectively. A "knob-like" mechanism could be implemented to offer additional tuning flexibility.

The DMA implements two multi-criteria ranking methods, one explicitly exploiting the ratio of accuracy and time via $AccD$ [9] and the other based on the idea of Data Envelopment Analysis [1, 6]. Technical details may be found in the aforementioned references. We do, however, point out here one of the advantages of using rankings rather than single model selection, especially when qualitative criteria are to be considered. There are indeed a number of qualitative criteria, such as comprehensibility, that practitioners may wish to take into account in their Data Mining activities (e.g., see [13]). Provided the set $L$ is diverse enough, ranking offers a simple way to add such selection criteria to the analysis *a posteriori*. For example, if comprehensibility is required, the user can simply go down the ranked list of algorithms produced by the DMA, ignoring those with black-box behavior, thus essentially constructing a new ranked list satisfying the additional criterion.

Finally, we briefly comment on the choice of $k$-NN as the meta-learner for the DMA. The main focus in the design of the DMA as a practical tool has been on simplicity and extensibility. As a lazy, inherently incremental learner, $k$-NN offers an effective solution to both. Anytime the DMA encounters a new task, it can not only make a recommendation on-line, based on its current meta-knowledge base, but also extend its knowledge base by automatically running the algorithms in $L$ off-line and storing the new task's corresponding ranking. Thus, the DMA is always usable, and as the number of tasks it handles increases, so does the accuracy of its predictions. However, as a lazy learner, $k$-NN does not compute an explicit meta-model and thus does not produce much insight regarding the meta-learning process. Other incremental learners could be considered, that offer both incrementality and explicit model building.

### 3.2 Functionality

The DMA uses a wizard-like interface to guide the user through the steps of the advice process: 1) dataset upload, 2) dataset characterization, and 3) ranking. For users who simply wish to test-drive the DMA, a simple demo example is available. The DMA expects datasets to be formatted according to C4.5's convention [25], with a few additional constraints. Given the prototypical nature of the DMA, this choice avoids the implementation of costly data format conversions. The DMA does however provide a simple tool to check files before upload.

Conscious of the confidential nature of many commercial datasets, the DMA provides three possible treatments of data. At the lowest level, all information is public, meaning that all users of the DMA have full access to the task's data. At the highest level, all information is private meaning that only the data owner may look at the data, generate rankings for it, run algorithms on it and use it as meta-data. The intermediate level offers a trade-off by making the base-level data private but the derived meta-data (i.e., the task's characterization) public. Thus, only the data owner may look at the data and run algorithms on it, but all users may generate rankings for it and use it as meta-data.

Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05)
0-7695-2495-8/05 $20.00 © 2005 IEEE

Once the data has been successfully uploaded, its characterization is computed. The user then selects the desired ranking method and trade-off between accuracy and time. Finally, the DMA computes and displays its proposed ranking for the new task. As a convenience for the user and a way to check the accuracy of the meta-learner, the DMA implements all of the base level algorithms. The user is able to select some or all of the algorithms for the DMA to run. Although the induced models themselves are not returned, the DMA reports 10-fold cross-validation accuracy, true rank and score, and, when relevant, training time. A simple example is shown in Figure 1, where some algorithms were selected for execution (the main selection criterion is accuracy in 1(a) and training time in 1(b)). Notice how in both cases, although the actual best is not always ranked first by the DMA, it is within the top 3 of the proposed ranking. This is consistent with most of our empirical findings, and is further evidence of the value of using rankings rather than single-best selection.

**Ranking table**

Dowload results | register now! | learn more about the online advisor

| Predicted Rank | Algorithm | Predicted Score | Status | Run | Accuracy | Time | True Rank | True Score |
|---|---|---|---|---|---|---|---|---|
| 1. | c50rules | 1.031 | finished | -- | 0.2830000 | ? | 6 | 1.003 |
| 2. | lindiscr | 1.03 | finished | -- | 0.2340000 | ? | 1 | 1.048 |
| 3. | c50tree | 1.026 | -- | | -- | -- | -- | -- |
| 4. | ltree | 1.023 | -- | | -- | -- | -- | -- |
| 5. | clemMLP | 1.017 | finished | -- | 0.2680000 | ? | 5 | 1.006 |
| 6. | c50boost | 1.017 | -- | | -- | -- | -- | -- |
| 7. | ripper | 1.009 | -- | | -- | -- | -- | -- |
| 8. | mlcnb | 1 | finished | -- | 0.2430000 | ? | 2 | 1.036 |
| 9. | clemRBFN | 0.948 | -- | | -- | -- | -- | -- |
| 10. | mlcib1 | 0.913 | finished | -- | 0.3180000 | ? | 10 | 0.938 |

(a) Emphasis on Accuracy

**Ranking table**

Dowload results | register now! | learn more about the online advisor

| Predicted Rank | Algorithm | Predicted Score | Status | Run | Accuracy | Time | True Rank | True Score |
|---|---|---|---|---|---|---|---|---|
| 1. | lindiscr | 1.153 | finished | -- | 0.2340000 | ? | 1 | 1.154 |
| 2. | c50tree | 1.144 | finished | -- | 0.2940000 | ? | 2 | 1.101 |
| 3. | c50rules | 1.07 | finished | -- | 0.2830000 | ? | 4 | 1.058 |
| 4. | ltree | 1.061 | -- | | -- | -- | -- | -- |
| 5. | mlcnb | 1.051 | -- | | -- | -- | -- | -- |
| 6. | c50boost | 1.009 | -- | | -- | -- | -- | -- |
| 7. | ripper | 1.002 | -- | | -- | -- | -- | -- |
| 8. | clemMLP | 0.909 | -- | | -- | -- | -- | -- |
| 9. | mlcib1 | 0.903 | finished | -- | 0.3180000 | ? | 8 | 0.931 |
| 10. | clemRBFN | 0.842 | -- | | -- | -- | -- | -- |

(b) Emphasis on Training Time

**Figure 1. Proposed and selected actual rankings for a sample task**

### 3.3 Usage Report

The DMA was officially launched around May 2002. Little has been done in terms of formal publicity for the DMA. Mention was made of it in the *ITtoolbox DataMining-Select* discussion group in October 2002 and in *KDnuggets News* in January 2003. Some recent introductory papers on meta-learning also contain references to it (e.g., see [14, 29]). Hence, most of the publicity for the DMA has been word-of-mouth and self-referral. We have been systematically tracking usage of the DMA site since its inception. Figure 4 shows the evolution of the number and type of registered users over time. All users directly connected to the METAL project have been excluded. Note that for reasons of relocation, the DMA site was inaccessible for 3 months between November 2004 and January 2005, causing the plateau effect observed over that period.

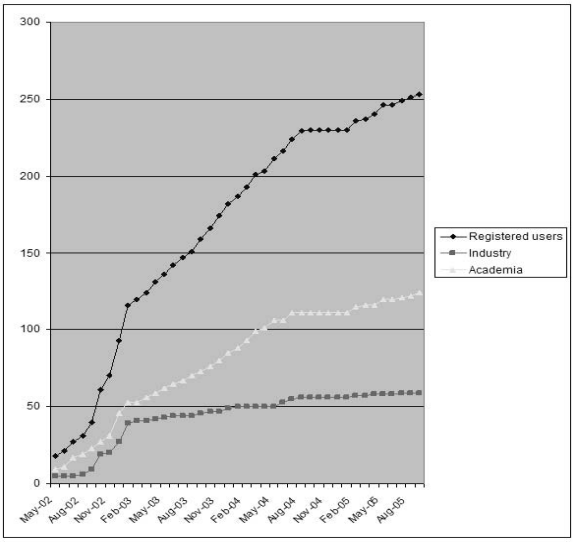


**Figure 2. Evolution of the Number of Registered DMA Users (May 2002 - Sep 2005)**

The total number of (reliable) registered users as of mid-September 2005 is 253, with about 25% coming from industry, 50% from academia, and 25% of unidentifiable origin, across the world (45 countries have been reliably identified). It is worth noting that these percentages have remained rather constant over time and are thus representative of the pattern of usage of the DMA so far.

The meta-knowledge base was initialized with about 67 classification tasks, mostly issued from the UCI repository [7]. An additional 83 tasks have since been uploaded, albeit from a rather limited number of individuals outside of the METAL project. As far as site traffic, the DMA site has been receiving about 10-15 visits per day on average over the past few months (down from 30-35 at the end of 2004, before the site was relocated).

The DMA site also offers an opportunity for users to

Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05)
0-7695-2495-8/05 $20.00 © 2005 IEEE

provide explicit feedback about the system by answering a small questionnaire. Although only 7 individuals have completed the questionnaire so far, making reliable statistical inference difficult, the results are indicative of the value of decision support systems for Data Mining practitioners. Most responders consider themselves experienced and declare Data Mining as part of their core business. 6 out of the 7 responders declare that: 1) they are somewhat satisfied with the DMA's advice (1 is very satisfied); 2) they would recommend the DMA to others; 3) they favor ranking over single-best selection (none prefer single-best selection); 4) they would find it a helpful service that algorithms may be run and results downloaded; and 5) accuracy and comprehensibility are the most important criteria in model selection (training time comes 4th after model complexity). Not only are these findings encouraging, they also confirm some of the DMA's design decisions (e.g., ranking, multi-criteria selection) and point in the direction of further useful developments (e.g., model download).

## 4 Conclusion

This paper revisits the meta-learning task for model selection in classification Data Mining. It discusses the main issues involved and practical implications, arguing that meta-learning provides a viable and robust mechanism for model selection. The Data Mining Advisor, a prototype web-enabled system for model selection with meta-learning, is briefly described and findings about its usage since its inception are reported. They seem to indicate the potential of such advice strategies in Data Mining activities.

A number of directions for future research and development may be followed. In terms of meta-learning for model selection, these include:

- Further work on task characterization is needed, to assess both relevance in meta-learning and insightfulness in delineating interesting tasks.
- Meta-learning is inherently incremental and thus raises a number of questions, such as how prior knowledge may be included or how the meta-learner adapts to new datasets, new base level algorithms, etc.
- Collecting training data at the meta-level is expensive and slow. Are there systematic ways of generating training data for meta-learning? The answer likely depends on the availability of a relevant characterization.

In terms of the Data Mining Advisor, it is hoped that usage will continue to increase and feedback provided so that the tool may be improved to reflect the needs and practices of users. The following are already considered:

- Allow users to download the models induced by selected algorithms in some standard format, such as PMML, so that these are immediately available for integration in business applications.
- Further work is needed on multi-criteria ranking. The DMA includes ony two quantitative ones so far (accuracy and training time), but evidence suggests that other more qualitative ones may be more relevant.
- The set of base level algorithms must be reviewed and refined. Although the current set provides some diversity, one may argue that the use of 3 versions of C5.0 may cause some redundancy and that the system may be better served by the inclusion of representatives of other model classes, such as Support Vector Machines.

Recent work in meta-learning has been encouraged by the early success of the DMA. Not only does the prototype offer a testbed for research results, it provides a simple means of obtaining users' feedback. In time, we hope to integrate DMA-like functionality into commercial Data Mining packages. As a first step, the Data Characterization Tool was implemented as a node for Clementine and an implementation of the DMA has been achieved in Weka [30].

## 5 Acknowledgments

The author, who formerly served as project coordinator, wishes to thank all of the members of the METAL project consortium: the University of Bristol, LIACC at the University of Porto, The University of Geneva, The Austrian Institute for Artificial Intelligence, DaimlerChrysler, SPSS (until 2000), and Dialogis (until 2001). Significant contributions were also made by The University of Applied Sciences Bonn-Rhein-Sieg, The Josef Stefan Institute and ELCA Informatique SA as associate partners. Special thanks are due to Carlos Soares and Pavel Brazdil for continuing to host and maintain the DMA's Web site.

## References

[1] P. Andersen and N. Petersen. A procedure for ranking efficient units in data envelopment analysis. *Management Science*, 39(10):1261–1264, 1993.

[2] H. Bensusan. Odd bites into bananas don't make you blind: Learning about simplicity and attribute addition. In *Proceedings of the ECML'98 Workshop on Upgrading Learning to the Meta-level: Model Selection and Data Transformation*, pages 30–42, 1998.

[3] H. Bensusan and C. Giraud-Carrier. Discovering task neighbourhoods through landmark learning performances. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000), LNAI 1910*, pages 325–330, 2000.

[4] H. Bensusan, C. Giraud-Carrier, and C. Kennedy. 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*, pages 109–118, 2000.

[5] H. Bensusan and A. Kalousis. Estimating the predictive accuracy of a classifier. In *Proceedings of the Twelfth European Conference on Machine Learning*, volume 2167 of *LNCS*, pages 25–36, 2001.

[6] H. Berrer, I. Paterson, and J. Keller. 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.

[7] C. Blake and C. Merz. UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Science, 1998.

[8] P. Brazdil and C. Soares. A comparison of ranking methods for classification algorithm selection. In *Proceedings of the Eleventh European Conference on Machine Learning*, pages 63–74, 2000.

[9] P. Brazdil, C. Soares, and J. Pinto da Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.

[10] Data mining advisor. http://www.metal-kdd.org.

[11] R. Engels and C. Theusinger. Using a data metric for offering preprocessing advice in data-mining applications. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, pages 430–434, 1998.

[12] J. Fuernkranz and J. Petrak. An evaluation of landmarking variants. In *Proceedings of the ECML/PKDD-01 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-learning*, pages 57–68, 2001.

[13] C. Giraud-Carrier. Beyond predictive accuracy: What? In *Proceedings of the ECML'98 Workshop on Upgrading Learning to the Meta-level: Model Selection and Data Transformation*, pages 78–85, 1998.

[14] C. Giraud-Carrier and J. Keller. Meta-learning. In J. Meij, editor, *Dealing with the Data Flood: Mining Data, text and Multimedia*, pages 832–844. STT/Beweton, The Hague, STT 65, 2002.

[15] M. Hilario and A. Kalousis. Fusion of meta-knowledge and meta-data for case-based model selection. In *Proceedings of the Fifth European Conference on Principles of Data Mining and Knowledge Discovery*, pages 180–191, 2001.

[16] R. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[17] A. Kalousis and M. Hilario. Model selection via meta-learning: A comparative study. In *Proceedings of the Twelfth International IEEE Conference on Tools with AI*, pages 406–413, 2000.

[18] E. Keogh and T. Folias. The UCR time series data mining archive [http://www.cs.ucr.edu/ eamonn/tsdma/index.html]. Riverside CA. University of California - Computer Science & Engineering Department, 2002.

[19] E. Keogh, S. Lonardi, and C. Ratanmahatana. Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215, 2004.

[20] G. Lindner and R. Studer. Ast: Support for algorithm selection with a cbr approach. In *Proceedings of the Third International Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 418–423, 1999.

[21] METAL: A meta-learning assistant for providing user support in machine learning and data mining. ESPRIT Framework IV - LTR Reactive Project Nr. 26.357, 1998-2001.

[22] D. Michie, D. Spiegelhalter, and C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[23] Y. Peng, P. Flach, P. Brazdil, and C. Soares. Improved data set characterisation for meta-learning. In *Proceedings of the Fifth International Conference on Discovery Science*, pages 141–152, 2002.

[24] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, pages 743–750, 2000.

[25] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1993.

[26] C. Schaffer. A conservation law for generalization performance. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 259–265, 1994.

[27] S. Sohn. Meta analysis of classification algorithms for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137–1144, 1999.

[28] M. van Someren. Model class selection and construction: Beyond the procustean approach to machine learning applications. In G. Paliouras, V. Karkaletsis, and C. Spyropoulos, editors, *Machine Learning and Its Applications: Advanced Lectures, LNCS 2049*, pages 196–217. Springer-Verlag, 2001.

[29] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares. Using meta-learning to support data-mining. *International Journal of Computer Science Applications*, I(1):31–45, 2004.

[30] Wekametal. http://www.cs.bris.ac.uk/∼farrand/wekametal.

[31] I. Witten and F. Eibe. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000.

[32] D. Wolpert. Any two learning algorithms are (almost) exactly identical. Technical report, NASA Ames Research Center, 2001.

[33] D. Wolpert. The supervised learning no-free-lunch theorems. In *Proceedings of the Sixth On-line World Conference on Soft Computing in Industrial Applications*, pages 325–330, 2001.

[34] D. Wolpert and W. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.