A Process Model for Developing Inductive Applications *

Floor Verdenius
AgroTechnological Research Institute (ATO-DLO)
Postbox 17, 6700 AA Wageningen, The Netherlands
e-mail: F.Verdenius@ato.dlo.nl

Robert Engels
Institute AIFB
University of Karlsruhe, Karlsruhe, Germany
e-mail: engels@aifb.uni-karlsruhe.de

August 5, 1997

Abstract

A growing interest in real-world applications of inductive techniques signifies the need for methodologies for applying them. So far a number of methodologies for applying inductive learning techniques are described. After reviewing several published approaches, a number of unsolved problems are discussed, two major problems being the lack of attention to non-technical issues and the focus of most approaches on specific, well defined problems with a limited scope. We propose the MEDIA-model as a reference structure for the application of inductive learning techniques that covers the issues mentioned in other approaches and generalises from problem specific approaches. The model is part of a methodology that aims at supporting the application of inductive learning techniques in various settings, and helps to plan projects where such techniques are involved.

1 Introduction

Machine Learning techniques become popular tools for solving real world problems. Recently several surveys in real world applications that exploit these techniques have been reported (Rudström, 1995) (Verdenius, 1997a). With the transition of these techniques from a research environment to the industry, the research focus shifts from technical issues towards the process of designing and implementing applications. In literature, this development is reflected in the growing number of reports on process models and methods for Machine Learning (ML) application (Kodratoff et al., 1994) (Brodley and Smyth, 1997) (Garner et al., 1995) and work on KDD application support ((Engels, 1996), (Craw et al., 1992), (Engels et al., 1997b)). The shift of focus is also reflected in a number of workshops on the process of ML application that were organised within recent ICML conferences (Langley and Kodratoff, 1993), (Aha and Riddle, 1995), (Engels et al., 1997a).

Much of the existing work on methodological aspects of ML application focuses on the technical details of applying the technique, thereby ignoring the higher level design aspects of ML application. As a result, potential appliers of ML techniques are only supported in solving their problem when they have already set major problem solving steps as solution design, data acquisition, data analysis and technique selection.

*Published in the: Proceedings of the 7th Seventh Belgian-Dutch Conference on Machine Learning BENELEARN-97, Tilburg, Oct 21st, 1997

This paper presents a model that structures the total process of ML application. The presented model starts at the level of problem statement, and structures all activities until the level of the individual techniques. The final goal of our work is to provide support, i.e. guidance and required tools, in the development of these applications.

The text is structured as follows. In the next section, we introduce the process of ML development, and discuss the relevant literature. Then, in section 3 major problems in the process of ML application are discussed, and potential solutions are evaluated. Section 4 discusses the *Me*thod for *D*esigning *I*nductive *A*pplications. Then, we discuss the model, partly on the basis of realised application projects. Section 6 concludes.

2 The Process of Applying Inductive Techniques

In recent years machine learning techniques have become mature. Techniques for standard tasks, such as classification and clustering, have been improved and refined. More complex tasks, such as cost sensitive classification, have been covered by newly developed techniques. As in many technical disciplines in their initial phase of development, the machine learning (ML) society has realised progress mainly by adding technical improvements and innovations to existing ML techniques. Methodological approaches on problem analyses and technique selection have received less attention. As a result, practitioners are nowadays equipped with a multitude of learning techniques, many of them being very specific for a problem type, data set characteristics and type of application. Moreover, for most tasks non-ML inductive techniques such as inductive statistics and neural networks are available that have a similar functionality as the available ML technique.

The main questions for a methodological approach are:

- How to integrate, from the design phase on, an ML solution within an embedding system (i.e. realising a learning function in a large complex software system, or integrating an adaptive knowledge base in a non-automated process)?
- How to use the available techniques to solve a real world problem?
- How to recognise, in an early stage of problem solving, application potential for ML techniques, and equipping the development process as adequate as possible?
- How to analyse tasks in order to optimally exploit this potential?
- How to select a good, if not the optimal, technique to solve a specific problem? And how to configure this technique for optimal performance?

Various authors have attended the process of ML application. (Weiss and Kulikowski, 1991) present for the classification task an approach with the knowledge acquisition aim of extracting a (knowledge) model from data. The application of the model, and integration in a system or solution is not covered by their approach. The requirement for their approach is to provide a model to be used for classification. In their approach, the problem is seen as the problem of selecting a suitable technique. The toolbox they present contains four (groups) of techniques which the authors find useful for acquiring such knowledge. The technique(s) are processed in order, until a technique is encountered that satisfies the requirements. Satisfaction is primarily assessed in terms of model accuracy. The order is a combination of increasing expressive power and interpretability of the resulting model, which serves as a secondary criteria for satisfaction: the best model performs accurate enough, and is as expressive and interpretable as possible.

(Kodratoff et al., 1994) and (Craw et al., 1992) provide an approach to complement the Machine Learning Toolbox that resulted from the ESPRIT MLT project. The toolbox contains about 40 inductive techniques for classification. The approach, designed for implementation in the MLT Consultant tool, was designed to support users in selecting the proper technique for their problem.

The approach structures application process in decision-tree shaped taxonomies. Taxonomies exist for items such as *ML application goals* (learning for: similarity detection, acquire knowledge, classify instances, ...), *Nature of available data* (examples availability: Incremental, Batch, ...), *Nature of available background knowledge* (background knowledge: Usable, Obliged to use, ...). Every decision structure is used to select a subset of potentially suitable techniques. Another workbench specific approach is found in (Garner et al., 1995). Similar to MLT, WEKA contains a number of classification techniques, and the main aim is to come up with a specific model. In comparison with the former approach, the WEKA approach is more process oriented and less knowledge oriented. Moreover, the functional environment of application of the ML technique is not static, but actively influenced *as part of the ML application process.* This is specifically the case for pre- and post-processing of data.

The approach of (Brodley and Smyth, 1997) is yet one step more general in that they extend the scope of their method for applying ML techniques to the process of analysing the problem environment, not only in terms of data, but also in terms of (what they call) domain specific factors such as application specific and human factors. In their description of their approach, the authors explicitly discuss the important aspects to be analysed. This method is, as the ones discussed before, specific for classification tasks.

Even more general is the approach for realising neural network applications as described by (DTI, 1994). In this approach the problem to solve is less specific then in the ones discussed before. Neural networks are suitable for a broad range of tasks, amongst others classification, optimisation, and prediction of continuous values. Furthermore, the scope of this approach, as well as its formulation, is much more detailed. The organisation of the total approach is similar to classical *waterfall approaches* for software development, with clearly indicated phases and mile stone products. Some of these phases are extremely detailed, others are more globally defined.

(Wirth et al., 1997) present a general approach for KDD process. In this approach, we can observe the same aspect of generality towards the task to perform and the independence of techniques to be used, as was observed for (Brodley and Smyth, 1997).

There are a number of characteristics that can be used to organise all these approaches:

Technique orientation Process models can be oriented towards (a group of) techniques, or they can be technique independent. An example of a technique group oriented approach is the one used in MLT Consultant. The total approach focuses on selecting one technique from a set of techniques. An example of an technique independent approach is that of (Brodley and Smyth, 1997). Here, a task is supported, without limitations on the specifying the techniques to be used.

Task orientation Process models can be oriented towards a specific (expert) task, or they can be unspecific towards the task to learn. Task specificity leads to two characteristics: the lacking of a task analysis phase, and the presence of task specific analysis tools. An example of a task specific approach is that of (Brodley and Smyth, 1995), being completely configured around the classification task. Examples of task independent approaches are described in (Engels, 1996) and (Verdenius, 1997b). In these approaches derivation of a task decomposition is a central notion in the process. It is noticed that in principle technique specificity implies task specificity.

Application orientation Process models can be oriented towards a specific application mode, or they can be mode independent. An example of mode specific approach can be found in (Engels, 1996). Here the intended usage is that of Knowledge Discovery in Databases (KDD). In KDD, the resulting knowledge model is more essential than the resulting system (if any). The latter is even more true for a more data analysis oriented approach (e.g. (Garner et al., 1995)). Examples of application mode independent process models are scarce. Later in this paper we will present such a process model.

3 Unsolved Problems

The existing approaches can be criticised on three main aspects:

Qualitative definition of development activities The approaches, at least in their published form, exist of enumerations of activities. The contents of the activities are qualitatively described, which also accounts for the inputs and outputs that connects these activities. Neither the activities, nor the input/output flows are unambiguously defined. As such, there is hardly any support for developers who are confronted with the need of applying inductive techniques in their applications.

Limited scope Most approaches either tend to support application of a specific technique(group) (e.g. MLT focusing to a set of ten (learning) algorithms (Consortium, 1993)), or to support the realisation of a specific task (e.g. (Brodley and Smyth, 1997), (Brodley and Smyth, 1997)), and often they combine these aspects (e.g. classification with ML- techniques). This implies that important aspects of an applications development cycle, being the design task and the technique selection task, are taken for granted. This means that such approaches do not support the initial stages of a project because of their bias to specific directions. However, the real temptation in applying inductive techniques lies in the guidance of the functional design in such a manner that inductive techniques, if appropriate, are applied for those tasks where they are optimally suited, while also be able to propose other solutions when appropriate. In other words, the developer needs to be supported in his design tasks in such a way that purposeful deployment of inductive techniques is the result of the project. In our point of view, none of the previously discussed approaches succeeds in fulfilling these goals.

Application specific Many approaches limit themselves towards a specific application type. An specific approach (Reinartz and Wirth, 1995), is meant to cover the KDD task. However, the KDD task is only a specific case of the much broader process of inductive technique application.

The approach of (Brodley and Smyth, 1997) is limited in several ways. First, it concentrates on classification tasks. As explained above, this is only one of a large family of tasks inductive techniques can be applied for. Second, it remains very general on important issues as *identification of inductive technique application opportunities*, *technique selection* and *technique configuration*.

Another approach (Garner et al., 1995) is limited to data acquisition and technique training, is limited to classification tasks and avoids technique selection by brute force assessment of training methods.

4 The MEDIA Model

The MEthod for the Development of Inductive Applications (*the* MEDIA *model*) represents a reference structure. That is, it is not meant for a specific type of application. Instead, it offers an open framework to describe the design of inductive applications. As such, the model describes a comprehensive overview of activities and the connecting information flows. The activities in the MEDIA model are not necessarily performed exhaustively. Based on the requirements for a specific application and the results of other project activities, certain activities may become obsolete. Thus the precise list of actions has to be compiled on the basis of the specific functional and non-functional requirements that the application/employer imposes on a project[1]. The MEDIA model is meant to offer support for the development of applications of inductive learning techniques. As such, it bears the ambition to overcome the limitations mentioned in the previous sections. Figure 1 presents the general structure of the method.

[1]In earlier versions of the model this aspect of dynamically defining the development path during the project, was explicitly located in an overall *project management activity* (cf. Methodology engineering).

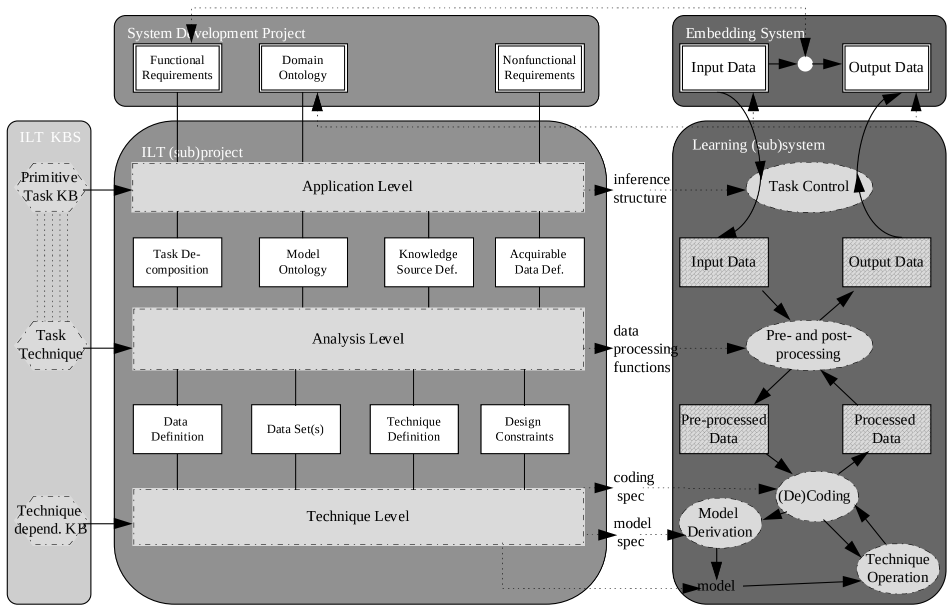

Figure 1: An overview of the MEDIA model

From the figure, several of the major properties of the MEDIA model can be learned. The figure consists of three sections. In the centre, the *activity* structure of the model is depicted. On the left side the Inductive Learning Technique Knowledge Base (ILT-KB) is found, whereas the right section represents the output subsystem, i.e. the system that in the end will represent and perform the mapping function f(x) as found in the Embedding System structure (right top hand side).

4.1 The Heart of the Model: the Activity Structure

As mentioned above, the heart of the MEDIA model is formed by the activity structure as depicted in the central section of figure 1.

The main activities performed in the MEDIA model are ordered in layers in the activity model (The central structure in figure 1). The structure on top of the activity model forms the interface between the ML application activities and the environment. The interface consists of *functional requirements*, *non-functional requirements* and the *domain ontology* of the embedding application. The functional requirements define the *operational task* that has to be realised in the learning application. In most cases, this operational task will be more complex as just a learning task. Mostly an ILT project is started with the goal to employ certain knowledge which should be implicitly present in the data. Making this knowledge explicit (e.g. in the form of a generated model) forms a subtask in the overall task decomposition. Non-functional requirements define the constraints the application has to satisfy (e.g. response time, memory resources, interpretability of the resulting model). The domain ontology defines the concepts as defined in the embedding application.

Next in hierarchy the *Application level* is found. On this level, several aspects of the applications operational task are analysed. This operational task is then refined in a:

- *task decomposition*, breaking down the task in subtasks, until a set of simple, formally described tasks is found. The model acquisition tasks are also part of this task decomposition, as far as inductive techniques are available (see (Engels, 1996)). Additionally, the task decomposition describes all steps that are necessary in order to perform the operational task.

Therefore, these tasks may comprise ordinary data processing tasks, such as pre- and/or post-processing.

- *Model ontology* that is a supplement to the domain ontology that adds tasks, relations and functions to the concepts as defined at the system development project structure.
- *knowledge source definition* that describe the models that are introduced in the task decomposition. On this level, models are described by function and content type. More formal aspects, such as representation and contents are defined at lower levels.
- *definition of acquirable data* that, at last, is an inventory of meta data (or data characteristics) describing the data that is available for induction.

On the *Analysis level* the link between the task decomposition and a set of techniques is established. For this purpose, characteristics of the domain, the operational data and the functional as well as the nonfunctional requirements are carefully analysed and interpreted (see also (Engels et al., 1997b)). This interpretation combines heuristic and formalisable aspects of learning techniques and their function. Output of this level includes:

- *data definition* defining the data items that are used for the inductive step,
- the accompanying *data sets* containing data for inducing models,
- the *technique definition* containing a high level description of technique design and lay out,
- and finally the *design constraints*. The latter are a direct translation of some of the non-functional requirements.

The last level is the *Technique level*. Here, technique specific aspects are defined. This includes parameter settings, model derivation and model operationalisation. Activity order is based on input/output relations between activities. For each activity the information that forms a necessary precondition of the activity in order to make the activity applicable is given.

4.2 Results of a Development Cycle

As discussed previously, activities are linked by shared input/output sets. Moreover, several activities produce final results, i.e. information items or products that are applied in the output structures. The output structures are found in the right section of figure 1. Each development/design level of the MEDIA model has a specific contribution to the final result: the application level defines the *task control* (which tasks have to be performed in what order to obtain the best result), the analysis level defines the techniques for pre- and post-processing, and at the technique level both model learning and model deployment is defined.

On the application level, a control flow is defined. A control flow defines the order of execution and defines iterations and their stop conditions in the subtasks that comprise the task decomposition. A good example is found in the preprocessing stage, where dimensionality reduction or value transformation often is a process with more iterations.

4.3 Tools for Development

The MEDIA model presupposes tools on the three development levels (application, analysis and technique level). Apart from standard software development tools and software methodology, learning systems require additional tools for defining inductive learning applications. On the *application level*, a task decomposition knowledge base facilitates the definition of task decompositions (Angele et al., 1996).

On the *Analysis level*, support focuses on the task-technique mapping problem. Here a knowledge base is used to link primitive tasks and techniques. The knowledge base is realized as a network structure that links primitive tasks to techniques, incorporating amongst other functional and non-functional information as defined at the application level (e.g. (van Someren et al., 1997) structures techniques concerning cost sensitive generalisation).

It is important to realise that the goal to be achieved in this phase of the project is not to identify the ultimate solution in terms of (for instance) accuracy. Instead, a complex (and in the current practice often implicitly defined) objective has to be satisfied. This objective combines functional requirements (often expressed in formalised or technical terms), such as accuracy, response time, training speed and memory usage, with *non-functional requirements* such as comprehensibility of representation, hard- and software platform, availability of technical expertise, personal preference of developers and clients, and availability of techniques. Satisfying these requirements will enable deployment of the model in practice.

On the *Technique Level*, optimisation of the technique for the specific tasks takes place. This means that parameter settings are adjusted where the context (defined by the non-functional requirements as well as the task decomposition) is taken into consideration. All tools are open, in the sense that new knowledge on inductive technique can easily be added to the system.

5 Discussion and further work

We can now compare the approaches of section 2 with the approach as expressed in the MEDIA model. The existing approaches were mostly characterised as either technique or task driven. We criticised this aspect because it ignored the needs and requirements as expressed by the client. An approach that is either technique or task driven will never be sufficient to provide an open unbiased methodology to an application developer. These approaches cannot fulfil the need for short (and predictable or controllable) development effort, since the limited scope of the discussed approaches only guarantees such a minimal development effort as long as one stays within this scope.

The MEDIA model is application oriented. It takes functional and non-functional requirements as its input, and delivers a system with the required functional behaviour, and is meant to be a general framework for the development of inductive learning applications. Induction in this approach is a useful tool for obtaining and representing knowledge that full-fills both the functional and non-functional requirements of an application. The MEDIA model in its initial formulation is an attempt to cover the needs from practice. It aims at filling the gap between research in ML and application of ML techniques, and facilitates practical application of ML in practice. Future research effort will concentrate on the following topics:

Model verification We are currently verifying the MEDIA model as presented here by analysing the development process of several projects. We hope to be able to asses the MEDIA models qualities as methodology for development of applications of inductive techniques. Our main concerns are:

- Can we identify all activities/information flows/products and knowledge bases postulated by the model in the several projects we analyse?
- Can we understand features/problems in the application process by identifying anomalies in process configuration as postulated in the MEDIA model?

Method development The current formulation of the MEDIA model is mainly descriptive. In the current state, methodological guidelines for designing and realising applications are scarce.

Tool development In the activity outline as depicted in the figure, knowledge bases are postulated on three levels. Implementation of these knowledge bases as specific tools seems appropriate. Especially the upper two levels, where the total process can be supported from one tool, is suitable for support within the MEDIA framework. On the technique level the central definition of tools is not feasible; a meta description of the functionality of a tool is needed, leaving technical details, as well as the development of the techniques as a research problem.

6 Conclusions

This paper provides an overview of the main approaches that tend towards a description of a methodology for the development of inductive applications. Nearly no general frameworks or methodologies are found, and therefore the problem arise that no general guidelines exist. Each approach is either biased to a certain task(group) or to a limited set of techniques, and therefore has a limited scope. This underlines the necessity of our approach, where no biases are defined concerning task groups and/or techniques that could be used. Our approach aims at the support from the early level onwards, and provides guidelines for the development of applications of inductive techniques in general. We are currently in the process of defining tools for that, as well as testing the approach against recent projects where inductive applications are build.

References

Aha, D. and Riddle, P. (1995). Working notes for applying machine learning in practice: A workshop at the twelfth international machine learning conference. Technical Report AIC-95-023, Naval Research Laboratory, Tahoe City, July 9th.

Angele, J., Fensel, D., and Studer, R. (1996). Domain and task modelling in mike. In Sutcliffe, A., van Assche, F., and Benyon, D., editors, *Domain Knowledge for Interactive System Design, Proceedings of the IFIP WG8.1/13.2 Joint Working Conference on Domain Knowledge for Interactive System Design.* Chapman & Hall.

Brodley, C. and Smyth, P. (1995). Applying classification algorithms in practice. In Aha, D., editor, *Proceedings of the Workshop on Applying Machine Learning in Practice at the ICML-95*, Tahoe City, CA.

Brodley, C. and Smyth, P. (1997). Applying classification algorithms in practice. *Journal of Statistics and Computing*.

Consortium, M. (1993). Final public report. Technical report. Esprit II Project 2154.

Craw, S., Sleeman, D., Granger, N., Rissakis, M., and Sharma, S. (1992). Consultant: Providing advice for the machine learning toolbox. In Bramer, M. and Milne, R., editors, *Research and Development in Expert Systems*, pages 5–23.

DTI (1994). *Neural Computing*. Learning Solutions, London.

Engels, R. (1996). Planning tasks for knowledge discovery in databases; performing task-oriented user-guidance. In Simounis, E., Han, J., and Fayyad, U., editors, *Proceedings of the 2nd Int. Conference on Knowledge Discovery in Databases*, pages 170–175, Portland, Oregon. AAAI-Press.

Engels, R., Evans, B., Herrmann, J., and Verdenius, F., editors (1997a). *Workshop on Machine Learning Application in the real world; Methodological Aspects and Implications (at the ICML-97)*, Nashville, TN, July 12th.

Engels, R., Lindner, G., and Studer, R. (1997b). A guided tour through the data mining jungle. In Uthurasamy, R., editor, *Proceedings of the 3rd International Conference on Knowledge Discovery in Databases*.

Garner, S., S.J.Cunningham, Holmes, G., Nevill-Manning, G., and Witten, I. (1995). Applying a machine learning workbench: Experience with agricultural databases. In Aha, D. and Riddle, P., editors, *Working Notes for Applying Machine Learning in Practice: A Workshop at the Twelvth International Machine Learning Conference.*, Washington, DC. Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Kodratoff, Y., Moustakis, V., and Graner, N. (1994). Can machine learning solve my problem? *Applied Artificial Intelligence*, 8:1–31.

Langley, P. and Kodratoff, Y., editors (1993). *Workshop on Real World Machine Learning Applications; ICML '93*, Amherst.

Reinartz, T. and Wirth, R. (1995). Towards a task model for kdd. In Kodratoff, Y., Nakhaeizadeh, G., and Taylor, C., editors, *Proceedings of the workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases. MLNet Familiarisation Workshop*, pages 19–24, Heraklion, Crete.

Rudström, A. (1995). Applications of machine learning. Technical Report 95-018, Department of Computer Science (DSV), University of Stockholm, Sweden. http:/www.dsv.su.se/ asa/getlic.html.

van Someren, M., Torres, C., and Verdenius, F. (1997). A systematic description of greedy optimisation algorithms for cost sensitive generalisation. In Cohen, P. and Lui, X., editors, *Proceedings of the 2nd Symposium on Intelligent Data Analysis*, London. Springer Verlag.

Verdenius, F. (1997a). Applications of inductive learning techniques: A survey in the netherlands. *AI communications*, 10(1).

Verdenius, F. (1997b). Developing an embedded neural network application: The making of the ptss. In Gielen, S. and Kappan, B., editors, *Proceedings of SNN'97: Europes Best Neural Networks Practice*, Amsterdam.

Weiss, S. and Kulikowski, C. (1991). *Computer Systems That Learn*. Morgan Kauffmann Publishers, Inc., San Mateo, California.

Wirth, R., Shearer, C., Grimmer, U., Reinartz, T., Schloesser, J., Breitner, C., Engels, R., and Lindner, G. (1997). Towards process-oriented tool support for kdd. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97)*.