Combining Induction Methods with the Multimethod Approach

Mitja Lenič
University of Maribor, FERI, Slovenia

Peter Kokol
University of Maribor, FERI, Slovenia

Petra Povalej
University of Maribor, FERI, Slovenia

Milan Zorman
University of Maribor, FERI, Slovenia

INTRODUCTION

The aggressive rate of growth of disk storage and, thus, the ability to store enormous quantities of data have far outpaced our ability to process and utilize that. This challenge has produced a phenomenon called data tombs—data is deposited to merely rest in peace, never to be accessed again. But the growing appreciation that data tombs represent missed opportunities in cases supporting scientific discovering, business exploitation, or complex decision making has awakened the growing commercial interest in knowledge discovery and data-mining techniques. That, in order, has stimulated new interest in the automatic knowledge induction from cases stored in large databases—a very important class of techniques in the data-mining field. With the variety of environments, it is almost impossible to develop a single-induction method that would fit all possible requirements. Thereafter, we constructed a new so-called multi-method approach, trying out some original solutions.

BACKGROUND

Through time, different approaches have evolved, such as symbolic approaches, computational learning theory, neural networks, and so forth. In our case, we focus on an induction process to find a way to extract generalized knowledge from observed cases (instances). That is accomplished by using inductive inference that is the process of moving from concrete instances to general model(s), where the goal is to learn how to extract knowledge from objects by analyzing a set of instances (already solved cases) whose classes are known. Instances are typically represented as attribute-value vectors. Learning input consists of a set of vectors/instances, each belonging to a known class, and the output consists of a mapping from attribute values to classes. This mapping hypothesis should accurately classify both the learning instances and the new unseen instances. The hypothesis hopefully represents generalized knowledge that is interesting for domain experts. The fundamental theory of learning is presented by Valiant (1984) and Auer (1995).

Single-Method Approaches

When comparing single approaches with different knowledge representations and different learning algorithms, there is no clear winner. Each method has its own advantages and some inherent limitations. Decision trees (Quinlan, 1993), for example, are easily understandable by a human and can be used even without a computer, but they have difficulties expressing complex nonlinear problems. On the other hand, connectivistic approaches that simulate cognitive abilities of the brain can extract complex relations, but solutions are not easily understandable to humans (only numbers of weights), and, therefore, as such, they are not directly usable for data mining. Evolutionary approaches to knowledge extraction are also a good alternative, because they are not inherently limited to a local solution (Goldberg, 1989) but are computationally expensive. There are many other approaches, like representation of the knowledge with rules, rough sets, case-based reasoning, support vector machines, different fuzzy methodologies, and ensemble methods (Dietterich, 2000).

Hybrid Approaches

Hybrid approaches rest on the assumption that only the synergetic combination of single models can unleash their full power. Each of the single methods has its advantages but also inherent limitations and disadvantages, which must be taken into account when using a particular method. For example, symbolic methods usually represent the knowledge in human readable form, and the connectivistc methods perform better in classification of unseen objects and are less affected by the noise in data as are symbolic methods. Therefore, the logical step is to combine both worlds to overcome the disadvantages and limitations of a single one.

In general, the hybrids can be divided according to the flow of knowledge into four categories (Iglesias, 1996):

- **Sequential Hybrid (Chain Processing):** The output of one method is an input to another method. For example, the neural net is trained with the training set to reduce noise.
- **Parallel Hybrid (Co-Processing):** Different methods are used to extract knowledge. In the next phase, some arbitration mechanism should be used to generate appropriate results.
- **External Hybrid (Meta Processing):** One method uses another external one. For example, meta decision trees (Todorovski & Dzeroski, 2000) that use neural nets in decision nodes to improve the classification results.
- **Embedded Hybrid (Sub-Processing):** One method is embedded in another. That is the most powerful hybrid, but the least modular one, because usually the methods are coupled tightly.

The hybrid systems are commonly static in structure and cannot change the order of how single methods are applied. To be able to use embedded hybrids of different internal knowledge representation, it is commonly required to transform one method representation into another. Some transformations are trivial, especially when converting from symbolic approaches. The problem is when the knowledge is not so clearly presented, like in a case of the neural network (McGarry, Wermter & MacIntyre, 2001; Zorman, Kokol & Podgorelec, 2000). The knowledge representation issue is very important in the multi-method approach, and we solved it in the original manner.

MULTI-METHOD APPROACH

Multi-method approach was introduced in Leni and Kokol (2002). While studying other approaches, we were inspired by the idea of hybrid approaches and evolutionary algorithms. Both approaches are very promising in achieving the goal to improve the quality of knowledge extraction and are not inherently limited to sub-optimal solutions. We also noticed that almost all attempts to combine different methods use the loose coupling approach. Of course, loose coupling is easier to implement, but methods work almost independently of each other, and, therefore, a lot of luck is needed to make them work as a team.

Opposed to the conventional hybrids described in the previous section, our idea is to dynamically combine and apply different methods in no predefined order to the same problem or decomposition of the problem. The main concern of the mutli-method approach is to find a way to enable a dynamic combination of methods to the somehow quasi-unified knowledge representation. In multiple, equally-qualitative solutions, like evolutionary algorithms (EA), each solution is obtained using an application of different methods with different parameters. Therefore, we introduce a population composed of individuals/solutions that have the common goal to improve their classification abilities on a given environment/problem. We also enable the coexistence of different types of knowledge representation in the same population. The most common knowledge representation models have to be standardized and strictly typed to support the applicability of different methods on individuals. Each induction method implementation uses its own internal knowledge representation that is not compatible with other methods that use the same type of knowledge. A typical example is WEKA, which uses at least four different knowledge representations for decision trees. Standardization, in general, brings greater modularity and interchangeability, but it has the following disadvantage: already existing methods cannot be directly integrated and have to be adjusted to the standardized representation.

Initial population of extracted knowledge is generated using different methods. In each generation, different operations appropriate for individual knowledge are applied to improve existing and create new intelligent systems. That enables incremental refinement of extracted knowledge, with different views on a given problem. The main problem is how to combine methods that