

# Agent-Based Knowledge Discovery

Winton H E Davies & Pete Edwards

Department of Computing Science  
King's College  
University of Aberdeen  
Aberdeen, AB9 2UE, UK.

{wdavies, pedwards}@csd.abdn.ac.uk

## Abstract

Agent-Based Knowledge Discovery provides a new technique for performing data-mining over distributed databases. By combining techniques from Distributed AI and Machine Learning, software agents equipped with learning algorithms mine local databases. These agents then co-operate to integrate the knowledge obtained, before presenting the results to the user. We are currently exploring the use of a new software agent language, Agent-K and the application of first order learning techniques to data-mining. However, the main area of investigation is how the agents should interact, and how the knowledge should be integrated.

## Introduction

This paper describes our current research<sup>1</sup> which spans the fields of knowledge discovery and software agents. Knowledge discovery (or data-mining) is concerned with extracting knowledge from databases and/or knowledge bases (Piatetsky-Shapiro & Frawley, 1991). Most data-mining systems employ one or more machine learning techniques to find previously unknown patterns in real world data. Later in this section we will briefly introduce the learning method we plan to use in our approach, and mention some general issues which differentiate data-mining from machine learning.

Traditionally, data-mining systems are designed to work on a single dataset. However, with the growth of networks, data is increasingly dispersed over many machines in many different geographical locations. In addition, databases are being joined by other sources of information that can be accessed over networks, e.g. knowledge bases, on-line dictionaries, etc. This has raised the issue of not only how to gather distributed information, but how new knowledge can be discovered in distributed information.

Software agents (Levy, Sagiv & Srivastava, 1994; Oates, Prasad & Lesser, 1994) are one response to the problem of using the vast amounts of information stored on networked systems. There are many types of software agent (Wooldridge & Jennings, 1994); however, agents are typically thought of as being 'intelligent' programs which have some degree of autonomy. We intend to design an open, flexible data-mining agent. A group of these agents will be able to co-operate to discover knowledge from distributed sources.

To date, most knowledge discovery systems have focused on extracting numeric or propositional knowledge from databases. For example, a propositional system could not learn the concept of *grandparenthood* from a database containing the names of people and their parents. Such concepts, together with recursive relations, are easily formulated as statements in first order predicate calculus. Our approach aims to find first-order relations in data, using techniques from Inductive Logic Programming (ILP). Many ILP algorithms allow background knowledge expressed in first order predicate calculus to be used during learning. Thus a knowledge base could be used to supply existing domain knowledge to an ILP-based data-mining agent.

Data-mining systems differ in certain ways from the machine learning algorithms which they are typically derived from. Firstly, they have to cope with large amounts of data. For example, learning over a census database containing information on millions of families is very different from looking at a few hand-crafted examples of 'model' families. The second problem is that real world data has a tendency to contain errors and missing information. Finally, a data-mining system aims to discover knowledge that is novel, useful, and understandable, which typically requires a human to focus the search and to provide feedback on the knowledge discovered.

Our high-level model is shown in Figure 1. One or more agents per network node are responsible for

---

<sup>1</sup> Support was provided through a UK Engineering & Physical Sciences Research Council (EPSRC) studentship.

examining and analysing a local data source. In addition, an agent may query a knowledge source for existing knowledge (such as rules or predicate definitions). The agents communicate with each other during the discovery process. This allows the agents to integrate the new knowledge they produce into a globally coherent theory. A user communicates with the agents via a user-interface. In addition, a supervisory agent, responsible for co-ordinating the discovery agents may exist. The interface allows the user to assign agents to data sources, and to allocate high level discovery goals. It allows the user to critique new knowledge discovered by the agents, and to direct the agents to new discovery goals, including ones that might make use of the new knowledge.

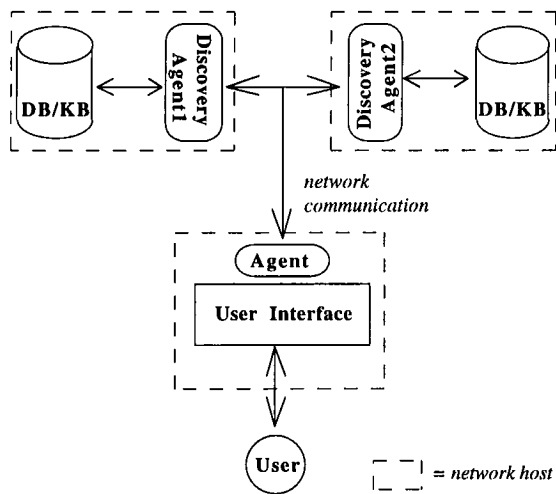


Figure 1: Data-Mining Using Multiple Agents

As far as possible, our intention is to base our work on the integration of existing technologies in the field of software agents and first order learning. This is in order to concentrate on the core issues of distributed data-mining. We intend to use agents based on Agent Oriented Programming (AOP) (Shoham, 1990), and the techniques developed as part of the Knowledge Sharing Effort (Patil et al., 1992). In addition, we have already identified a number of recent ILP algorithms, with which we plan to experiment. These include: the information-gain based FOCL (Pazzani & Kibler, 1992), CLAUDIEN (DeRaedt & Bruynooghe, 1993) and SIERES (Wirth & O'Rorke, 1992).

## Related Work

**Multi-Agent Learning** A number of co-operative distributed learning systems have been produced. MALE (Sian, 1991) is a homogeneous, blackboard-based system.

Each agent has a data-source and a clustering algorithm. The agents propose rules which characterise the data and critique other agents' proposals. Eventually a consensus about the knowledge extracted from the data is reached. ANIMALS (Edwards & Davies, 1993) is a heterogeneous multi-agent learning system. Each agent has local knowledge and either an inductive or deductive learning algorithm. Agents attempt to solve a problem-solving task by either retrieving the knowledge required, or by using learning to acquire it. Failures result in communication with other agents which are passed sub-goals, which are then treated as tasks. Both MALE and ANIMALS used propositional learning methods.

**First Order Knowledge Discovery** Some ILP systems have been applied to data-mining. One example is ENIGMA (Bergadano, Giordana & Saitta, 1991), which learnt fault diagnosis rules based on mechanical vibration data. Another is GOLEM (Muggleton, King & Sternberg, 1992), which learnt rules that predicted structural features in new proteins from existing protein data.

**Multi-Agent Knowledge Discovery** The Carnot Project (Woelk et al., 1992) addresses the problem of logically unifying distributed, heterogeneous business information. It appears that the underlying architecture uses software agents. Carnot provides a knowledge discovery system, presumably as an agent. However, we are uncertain whether this agent co-operates with similar agents, and are unsure as to the exact nature of the learning algorithm used.

**Distributed Database Mining** One approach which has emerged for mining distributed databases is to use a distributed database manager to provide seamless integration of the distributed data to data-mining algorithms (Simoudis, 1994). Our approach differs in that communication traffic between agents is restricted to the exchange of knowledge.

## Preliminary Work

To date we have tackled the following issues: the nature of the agent architecture; and the possible interactions between the agents during the data-mining process.

## Agent-K

The first step in designing our agent was to modify Agent-0 (Shoham, 1990) to use KQML (Finin et al., 1993) communication performatives. This is detailed in Davies & Edwards (1994). We hope to extend Agent-K to use the Knowledge Interchange Format and Ontolingua ontologies (Patil et al., 1992). This would allow our discovery agents to share knowledge with other KSE-based agents. We also plan to replace the basic Agent-0

interpreter with a variation on PLACA (Thomas, 1993), which would provide agents with a planning capability. Agents would thus have a means to plan interaction amongst themselves and with other KSE agents. We believe that Agent-K has already been a success, as it has demonstrated that the AOP approach is compatible with that of KQML. It has also provided a platform that we can eventually use for data-mining.

Agent-K provides a simple production rule mechanism that is used to program agents. The rules respond to incoming KQML messages and the current state of the agent, and if triggered, undertake a given action. In order to use the agents to support data-mining, it will be desirable to provide a generalised set of learning actions, which are independent of any specific learning algorithm. This set will have to include actions for negotiation between agents, about learning results.

## Knowledge Integration in Distributed Data-Mining

Individual agents will produce new knowledge based on their discovery goals, and the view of the distributed data. This knowledge will have to be integrated, so that it accounts for all views of the data.

Theory refinement and knowledge integration are related techniques. Theory refinement involves revising a theory with respect to new training examples. Knowledge integration involves combining two theories into a single unified theory. However, the learning techniques used for both are similar, and ILP algorithms in particular appear to make little distinction between revising clauses in response to new examples, and combining two sets of clauses and then revising them to fit the existing examples (Pazzani & Kibler, 1992).

Our initial decision to use an ILP learning algorithm was based on the insight that many ILP algorithms provide support for theory revision and knowledge integration. An agent based on such an algorithm could be used to both induce and integrate knowledge.

However, there is far more to consider than simply choosing an appropriate algorithm. Firstly, we must consider the nature of the examples (data) and discovery goals given to each agent. Then we must consider when the agents should co-operate; either before, during or after learning. Finally, if we decide that the co-operation should take place after learning, we have to decide how the agents will integrate the set of local results in order to reflect a global solution to the data-mining goal.

**Heterogeneous vs. Homogeneous Data-Mining** If each agent in the system is associated with a single database, then there are two basic types of interaction to consider. If each agent has the same discovery goal, and the same

database schema (though normally containing different values), then we refer to this as homogeneous data-mining. In this case the problem for the agents is to resolve partial results based on each partial view of the entire data.

If each agent has a different database and discovery goal, then the agents may use theories found by other agents as sub-theories. For example, if one agent learns a definition of *parent*, then a second agent might use this in its definition of *grandparent*. We refer to this as heterogeneous data-mining.

**Distributed Learning** There are three ways learning can occur when data is distributed. These relate to when agents communicate with respect to the learning process. As mentioned above, they can communicate before, during or after learning.

The first approach gathers the data into one place. The use of distributed database management systems to provide a single set of data to an algorithm is one example of this.

The second approach is for agents to exchange information whilst learning on local data. This is the approach taken by Sian (1991). No integration step is needed, as the agents are effectively working as a parallel algorithm over the dataset. This restricts the agents to using a single, highly specialised learning algorithm.

The third approach is for the agents to learn locally, and then to share their results, which are then modified by other agents in light of their own data and knowledge. This allows each agent to use a different algorithm if required. However, it raises the question of how all the local results should be integrated.

**Knowledge Integration** If the latter approach is adopted, then the local theories have to be integrated. It must be remembered that each agent's local results are correct for that agent's view of the data. Thus the fundamental problem is to compare local theories with previously unseen data, i.e. other agent's data. This data is of course summarised by the results produced by the other agents. There appear to be three alternatives for producing a single, global result.

The first approach involves a supervisor agent, in which one agent attempts to integrate all the local theories. However, this may lead to the transmission of large amounts of the original data, in order to test the accuracy of the new knowledge.

The second is a democratic version of the supervisor approach, with the agents working as a team to integrate their local results.

The third approach involves each agent taking other agent's theories and integrating them locally.

As can be seen from this discussion, the question of knowledge integration is a fundamental one in creating a distributed data-mining architecture.

## Conclusions

This paper describes our work to date on an agent-based approach to distributed knowledge discovery. We are currently investigating the application of ILP algorithms to data-mining tasks, and plan to adapt the selected ILP algorithm to work as an agent. We still have to decide how the agents will co-operate, and must produce an interface that will allow a user to interact with the agents.

A considerable number of issues have been raised during this work. For example, if agents interact during learning, some agents may be forced to wait while agents are still performing discovery. If agents interact upon completion of learning, knowledge integration may be computationally intensive. Another question is how agents should be selected to work on a given discovery goal. This might be addressed by the work on Site Description Languages (Levy, Sagiv & Srivastava, 1994).

Our long term goal is that agent-based knowledge discovery will allow distributed databases to be mined in a distributed manner: maximising the usage of distributed computing resources, and minimising network traffic.

## References

- F. Bergadano, A. Giordana & L. Saitta, Integrated Learning in a Real Domain, in *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro & W.J. Frawley (Eds.), MIT Press, 1991, 277-288.
- W. Davies & P. Edwards, Agent-K: An Integration of AOP and KQML, in *Proceedings of the CIKM'94 Intelligent Information Agents Workshop*, Y. Labrou & T. Finin (Eds.), 1994.
- L. DeRaedt & M. Bruynooghe, A Theory of Clausal Discovery, in *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI93)*, R. Bajcsy (Ed.), Morgan Kaufmann, 1993, 1030-1036.
- P. Edwards & W. Davies, A Heterogeneous Multi-Agent Learning System, in *1993 Proceedings of the Special Interest Group on Co-operating Knowledge Based Systems*, S. M. Deen (Ed.), University of Keele, 1993, 163-184.
- T. Finin et al., DRAFT Specification of the KQML Agent-Communication Language, 1993, *unpublished draft*.
- A. Y. Levy, Y. Sagiv & D. Srivastava, Towards Efficient Information Gathering Agents, in *Papers from the AAAI Spring Symposium on Software Agents*, Technical Report SS-94-03, AAAI Press, 1994, 64-70.
- S. Muggleton, R. King & M. Sternberg, Protein Secondary Structure Prediction using Logic, in *Proceedings of the Second International Workshop on Inductive Logic Programming*, ICOT, Tokyo, Japan, 1992.
- T. Oates, M. V. N. Prasad & V. R. Lesser, Co-operative Information Gathering: A Distributed Problem Solving Approach, *Technical Report Computer Science 94-66*, University of Massachusetts, 1994.
- R.S. Patil et al., The DARPA Knowledge Sharing Effort: Progress Report, in *Proceedings of KA92 - The Annual International Conference on Knowledge Acquisition*, Cambridge, MA, 1992.
- M. Pazzani & D. Kibler, The Utility of Knowledge in Inductive Learning, *Machine Learning*, 9, 1992, 57-94.
- G. Piatetsky-Shapiro & W. J. Frawley, *Knowledge Discovery in Databases*, MIT Press, 1991.
- Y. Shoham, Agent-Oriented Programming, *Technical Report STAN-CS-90-1335*, Stanford University, 1990.
- S. Sian, Extending Learning to Multiple Agents: Issues and a Model for Multi-Agent Machine Learning (MA-ML), in *Proceedings of the European Working Session on Learning - EWSL91*, Y. Kodratoff (Ed.), Springer-Verlag, 1991, 458-472.
- E. Simoudis, *personal communication*, 1994.
- S.R. Thomas, *PLACA, an Agent Oriented Programming Language*, Ph.D. Dissertation, Computer Science Department, Stanford University, 1993.
- R. Wirth & P. O'Rorke, Constraints for Predicate Invention, in *Inductive Logic Programming*, S. Muggleton (Ed.), Academic Press, 1992, 299-318.
- D. Woelk, W. M. Shen, M. Huhns & P. Cannata, Model Driven Enterprise Information Management in Carnot, in *Enterprise Integration Modelling: Proceedings of the First International Conference*, MIT Press, 1992.
- M. Wooldridge & N. R. Jennings, Intelligent Agents: Theory and Practice, *submitted to Knowledge Engineering Review*.