The WoRLD: Knowledge Discovery from Multiple Distributed Databases

John M. Aronis
University of Pittsburgh
aronis@cs.pitt.edu

Venkateswarlu Kolluri
University of Pittsburgh
venkat@lis.pitt.edu

Foster J. Provost
NYNEX Science and Technology
foster@nynexst.com

Bruce G. Buchanan
University of Pittsburgh
buchanan@cs.pitt.edu

Abstract

Inductive machine learning offers techniques for discovering new knowledge from business, medical, and scientific databases. Most techniques assume that all the relevant information for discovery has been gathered and assembled into a single table or database. With multiple databases it is possible to combine features from several perspectives and thus move beyond the confines of an ontology that was fixed by the designers of a single database. We introduce WoRLD ("Worldwide Relational Learning Daemon"), a system that uses spreading activation to enable inductive learning from multiple tables in multiple databases spread across the network. We describe the paradigm and the system, provide demonstrations on synthetic data sets, and then replicate two real-world successes of automated discovery.

1 INTRODUCTION

Inductive machine learning offers methods for discovering new knowledge from business, medical, and scientific databases. Although the need to learn across multiple tables has been realized [17], most inductive learning and data mining techniques assume that all the relevant information for discovery has been gathered and assembled into a single table or database. With multiple tables and multiple databases it is possible to combine features from several perspectives and thus move beyond the confines of an ontology that was fixed by the designers of a single database. In addition, it is highly likely that interesting and novel relationships exist in databases other than the one that motivated an inquiry in the first place.

Unfortunately, combining multiple databases requires substantial knowledge engineering to coalesce relevant information from multiple tables and databases. This requires a domain expert to determine which databases and which fields within them are relevant. Combining multiple databases also creates scaling problems for discovery programs. Even recent work that has begun to address the problem of learning across multiple databases [17] requires that the databases reside on the same machine. In practice, useful databases may exist in remote locations in an organization, or across the Internet, and in either case, their existence may not be known to the persons initiating an inquiry.

The WoRLD system described in this paper is an inductive rule-learning program that can learn from multiple databases distributed about the network. It demonstrates how databases codified for other purposes can introduce an open-endedness to the framework within which an induction program operates. Our long-term goal is to reduce the effort needed for knowledge discovery from databases by designing inductive systems that can locate new databases that contain information relevant to the current discovery task, and can utilize the information seamlessly for augmenting the inductive process.

2 DISTRIBUTED LEARNING

The key to WoRLD's ability to treat multiple databases transparently is its use of spreading activation [15], instead of item-by-item matching, as the basic operation of the inductive engine. This methods works by first labelling each item with a marker, here either "+" or "−" (for items in the concept of interest and its complement), then propagating these markers through databases looking for values where positive or negative markers accumulate. This process can span several databases, possibly on different machines.

2.1 Spreading Activation

Rule learning programs have contributed to real-world discoveries starting with the early work on MetaDENDRAL [2] and continuing through recent work (*e.g.*, in chemical carcinogenicity [11] and botanical toxicology [9, 10]). The fundamental step of these top-down inductive learners is generating and evaluating all the specializations of a partial rule. That is, given a partial rule of the form $C_1 \& \ldots \& C_n \rightarrow$ Concept, a learner tries to find conjuncts that can be added to the left-hand side of the rule to improve it.

People		
Name	*City*	*Car*
Sam	Pittsburgh	Corvette
John	Pittsburgh	Cutlass
Bob	New Castle	BMW
Tim	Los Angeles	BMW
Mary	New York	Jaguar

Figure 1: A Simple Database.

Consider the simple database shown in Figure 1. Suppose we want to characterize the set consisting of Sam, John, and Bob. Typical top-down inductive learners, such as MetaDENDRAL-style learners [2, 3, 18, 19], or decision-tree learners [16], start with the null rule (which covers everything) and generate additional conjuncts to specialize it, such as City = Pittsburgh, Car = Corvette, City = New Castle, etc. Each conjunct is matched against the data, and statistics are gathered. The statistics are fed to an evaluation function that decides which conjuncts should be further specialized on the next iteration.

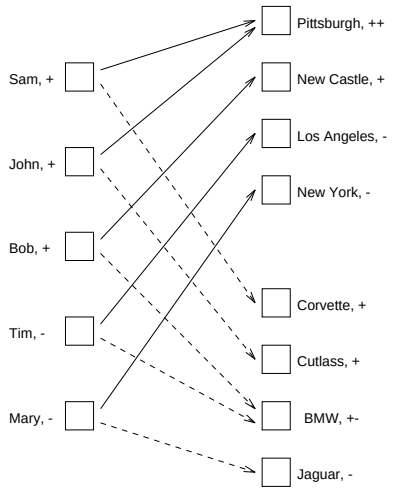

Figure 2: An Equivalent Simple Network.

WoRLD's inductive algorithm is identical to that of the MetaDENDRAL-style rule learners, specifically the RL program [3], except that the matching portion of the algorithm is replaced by spreading activation, and the collection and counting of markers. Consider the representation in Figure 2, in which attribute values are represented by pointers into the space of values (with a different type pointer for each different attribute). Instead of testing all the possibilities to characterize the set, WoRLD places a class marker on each data item. After propagating these markers along attribute links the learner then checks the coverages of predicates by counting how many positive and negative markers accumulate on the corresponding value nodes, thereby replacing the matching step in the rule learners.

People		
Name	*City*	*Car*
Sam	Pittsburgh	Corvette
John	Pittsburgh	Cutlass
Bob	New Castle	BMW
Tim	Los Angeles	BMW
Mary	New York	Jaguar

Cities	
City	*State*
Pittsburgh	Pennsylvania
New Castle	Pennsylvania
Los Angeles	California
New York	New York

Figure 3: A Two Table Relational Database.

2.2 Joining Databases

Now, consider the two-table database shown in Figure 3. In the traditional approach, a domain engineer who thought that *State* might be a relevant attribute for the discovery problem would *join* the tables on the column *City*, adding additional information to the data set. Subsequently, a standard learner can find that State = Pennsylvania perfectly characterizes the concept.

There are several problems with constructing joins in advance, although it is an obvious first suggestion. First of all, constructing all possible joins to create the *universal relation* will require a huge amount of space, particularly if the newly joined tables can also be connected to many others. Second, the resulting space of predicates to test will also be huge since they cover the space of all values in every field. Third, in a distributed environment the target tables may be maintained only as network pointers—

and these remote databases may also point to other foreign databases. Finally, and perhaps most importantly, the process of creating relevant joins manually is time- and effort-consuming, prone to errors of omission, subject to bias errors of commission and, for all the effort, still fixed in advance.

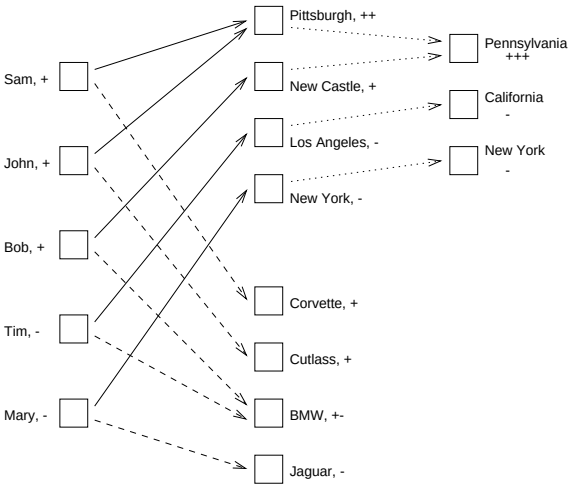

Figure 4: An Equivalent Two Table Network.

By using the spreading-activation method, there is no need to create a single join. Consider the alternate conceptual representation in Figure 4. Here, both tables are represented as a common linked structure. Markers are propagated across multiple links, when possible. The diagram shows the result of this process: all three positive markers have accumulated on Pennsylvania, the *unifying concept* of the original set. Thus, the basic learning operation—searching for predicates that have encouraging statistics with respect to their coverage of positive and negative examples—can be performed across multiple databases in a straight-forward manner.

In the WoRLD system, the operations of marker propagation and accumulation were implemented using basic database operations—relational algebra and common aggregate operations. It propagates markers from one database to another using a *join* operation to add relevant columns to the new table, along with their positive and negative counts. Accumulations of markers are counted by projecting on each column, then counting the markers for each value in that column using a summation operator. This design choice was made to facilitate learning across existing databases, each with its own query engine. Not unexpectedly, we have found that a direct implementation of marker propagation is much more efficient, but may require database format conversions.

An important feature of our approach is that there is no assumption that the databases all reside on the same machine. After markers are spread through the columns of a database they are collected and tallied, then sent to other databases to look for additional connections deeper in the relational structure. In our implementation the markers are propagated between databases as a stream of data. This stream can be transmitted across a regular network link, and the process is continued seamlessly on the next machine. Each database has additional information for each column specifying where on the network (or local machine) to look for potential joins, similar to links on the World-Wide-Web. As with the WWW, there is no need for a master map of the entire structure—each database has its own links into the network, and these can be followed as they are encountered.

2.3 Extensions

There are several natural extensions to WoRLD's basic process. First, we are not limited to relational databases. The spreading-activation learning methodology accomodates hierarchical background knowledge naturally. Classifications of objects can be represented using ISA links. Markers are propagated across these also, and unifying concepts can consist of classes as well as individual values. Second, ISA hierarchies can also include more complex inheritance structures. In [1], we describe a system that accomodates ISA hierarchies, role structures, and non-monotonic inheritance.

Finally, there is no limit on the number of joins that the method will accomodate. Markers are passed from one database to another based on potentially relevant relationships (those that would cause a domain engineer to consider a join). The markers are filtered by the distribution of values in the new column; they are tallied (as before), and they are passed to other databases that can be joined to the new column. The process continues as long as additional databases are found which can be joined. Although limits may be placed on the extent of spreading, for example by limiting the number of joins or the strength of semantic links, the horizon effect of stopping with arbitrary limits may be overcome by continuing the joins as long as the pathway looks promising, which can be judged by the number and distribution of markers on values.

3 EXPERIMENTS

We designed a class of synthetic data sets to evaluate the WoRLD system. Our goal was to demonstrate that the system could discover rules that re-

quired linking multiple databases across machines, to support our contention that coalescing multiple databases onto a single machine (into a single table) is not necessary in order for a standard rule-learning approach to be effective. To this end, we distributed the relevant features of concepts to be discovered across multiple tables, on both monolithic and distributed platforms. Figure 5 and Figure 6 show how seven tables were structured relationally for our discovery problems. The links represent joins. In each case, the seven tables were first stored on a single machine, then on seven machines on the local network. WoRLD successfully discovered the target concepts in every case.[1] Incidentally, there was a factor of four speedup from parallelism in each case. We consider this result to be encouraging; however, our primary thrust in the design of the prototype was not run-time efficiency.

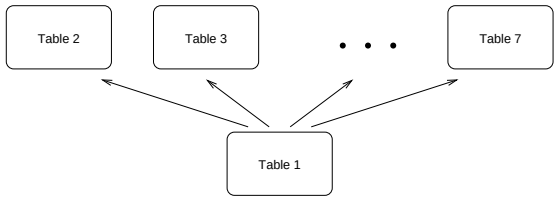

Figure 5: A Simple Relational Topology.

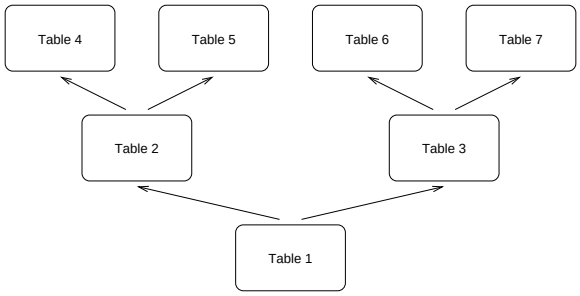

Figure 6: A More Complex Relational Topology.

As a further demonstration, we replicated the results of automated knowledge discovery from two real-world domains. In the first experiment we reproduced a clinically relevant botanical toxicology discovery made by KBRL [9], a precursor of WoRLD that learns across complex inheritance networks. In that work, we linked a database of poison plant exposures to databases of geographical and climate data in order to explain an interesting class of poisonings in terms of basic environmental principles. WoRLD was able to replicate KBRL's discovery without the need to manually link the databases into a single inheritance network. In the second replication, WoRLD successfully reproduced previously known concept definitions that identify high-risk pneumonia patients [4].

[1] Remember that WoRLD is functionally equivalent to a MetaDENDRAL-style rule learner in the learning biases it can incorporate.

4 CONCLUSIONS

The system described in this paper allows database maintainers to link their databases to others on the network for learning. The WoRLD system not only uses local links to foreign databases, but can also follow a series of links through databases on the network. But to achieve the long-range goal of autonomous learning across databases distributed around the World-Wide-Web, several significant problems will need to be solved.

The most significant problems of the WoRLD system are its reliance on a set of manually constructed and maintained links between databases, and its assumption of a standardized vocabulary across heterogeneous databases. One possible solution to these problems is to create a thesaurus, or ontology, of databases and terms in a domain. Similar suggestions have been made for molecular biology [8, 13], and medicine [14]. Instead of requiring database maintainers to specify explicit links to other databases on the network, appropriate links for a discovery program can be inferred from the ontology. Database maintainers would need to draw their concepts and terminology from this database, but many of the problems of reusability, synonyms, morphological variants, etc., will be solved. Considerable work has already been done in this direction [5, 6, 7, 12].

5 ACKNOWLEDGMENTS

This research was supported in part by National Science Foundation grant IRI-9412549, National Institutes of Health grant 2-P41-RR06009-06 (administered through the Pittsburgh Supercomputing Center), and the W.M. Keck Foundation.

References

[1] Aronis, J.; Provost, F.; Buchanan, B. 1996. Exploiting background knowledge in automated discovery. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

[2] Buchanan, B.; Mitchell, T. 1978. Model-directed learning of production rules. In *Pattern Directed*

Inference Systems, edited by D. Waterman and F. Hayes-Roth. Academic Press.

[3] Clearwater, S.; Provost, F. 1990. RL4: a tool for knowledge-based induction. In *Proceedings of the Second International IEEE Conference on Tools for Artificial Intelligence*. IEEE C.S. Press.

[4] Cooper, G.; Aliferis, C.; Ambrosino, R.; Aronis, J.; Buchanan, B.; Caruana, R.; Fine, M.; Glymour, C.; Gordon, G.; Hanusa, B.; Janosky, J.; Meek, C.; Mitchell, T.; Richarson, T.; Spirtes, P. 1997. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9, pp. 107–138.

[5] Fikes, R.; Cutkosky, M.; Gruber, T.; van Baalen, J. 1991. Knowledge sharing technology project overview. Technical Report KSL 91-71, Stanford University, Knowledge Systems Laboratory.

[6] Genesereth, M. 1991. Knowledge interchange format. *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Converence*. Morgan Kaufmann.

[7] Gruber, T. 1993. Ontolingua: a mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory.

[8] Karp, P. 1995. A strategy for database interoperation, *Journal of Computational Biology*, 2, pp. 573–586.

[9] Krenzelok, E.; Jacobsen, T.; Aronis, J. 1995. Jimsonweed (datura stramonium) poisoning and abuse ... an analysis of 1,458 cases. Abstracted in *Journal of Toxicology—Clinical Toxicology*, 33, p. 500.

[10] Krenzelok, E.; Provost, F.; Jacobsen, T.; Aronis, J.; Buchanan, B. 1995. Assessing patient referral patterns to a health care facility in plant exposure patients using computer artificial intelligence. In *European Association of Poisons Centers and Clinical Toxicologists Scientific Meeting.*

[11] Lee, Y.; Rosenkranz, H.; Buchanan, B.; Mattison, D.; Klopman, G. 1995. Learning rules to predict rodent carcinogenicity of non-genotoxic compounds. *Journal of Mutation Research*, in press.

[12] Lenat, D.; Guha, R. 1990. *Building Large Knowledge Based Systems.* Addison-Wesley.

[13] Markowitz, V. 1995. Heterogeneous molecular biology databases, *Journal of Computational Biology*, 2, pp. 537–538.

[14] McCray, A.; Razi, A. 1995. The UMLS knowledge source server. *Proceedings of MEDINFO 95.*

[15] Quillian, R. 1968. Semantic memory. In *Semantic Information Processing*, edited by M. Minsky. MIT Press.

[16] Quinlan, J. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

[17] Ribeiro, J.; Kaufman, K.; Kerschberg, L. 1995. Knowledge discovery from multiple databases. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining.* AAAI Press.

[18] Segal, R.; Etzioni, O. 1994. Learning decision lists using homogeneous rules. *Proceedings of the Twelfth National Conference on Artificial Intelligence.* AAAI Press.

[19] Webb, G. 1995. OPUS: an efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3, pp. 383–417.