Submitted to *Data Mining: Next Generation Challenges and Future Directions*,
H. Kargupta and A. Joshi (eds.), by AAAI/MIT Press

Relational Data Mining with Inductive Logic Programming for Link Discovery

Raymond J. Mooney, *Prem Melville*, *Lappoon Rupert Tang*, *Jude Shavlik*‡, *Inês de Castro Dutra*‡, *David Page*‡, *Vítor Santos Costa*‡

*Department of Computer Sciences
University of Texas
Austin, TX 78712-1188
{mooney,melville,rupert}@cs.utexas.edu

‡Department of Biostatistics and Medical Informatics and
Department of Computer Sciences
University of Wisconsin
Madison, WI 53706-1685
{shavlik,dpage}@cs.wisc.edu, {dutra,vitor}@biostat.wisc.edu

Abstract:
Link discovery (LD) is an important task in data mining for counter-terrorism and is the focus of DARPA's Evidence Extraction and Link Discovery (EELD) research program. Link discovery concerns the identification of complex relational patterns that indicate potentially threatening activities in large amounts of relational data. Most data-mining methods assume data is in the form of a feature-vector (a single relational table) and cannot handle multi-relational data. *Inductive logic programming* is a form of relational data mining that discovers rules in first-order logic from multi-relational data. This paper discusses the application of ILP to learning patterns for link discovery.

Keywords: ILP, Aleph, mFOIL, counter-terrorism, ensembles, EELD

1 Introduction

Since the events of September 11, 2001, the development of information technology that could aid intelligence agencies in their efforts to detect and prevent terrorism has become an important focus of attention. The Evidence Extraction and Link Discovery (EELD) program of the Defense Advanced Research Projects Agency (DARPA) is one research project that attempts to address this issue. The establishment of the EELD program for developing advanced software for aiding the detection of terrorist activity pre-dates the events of 9/11. The program had its genesis at a preliminary DARPA planning meeting held at Carnegie Mellon University after the opening of the Center for Automated Learning and Discovery in June of 1998. This meeting discussed the possible formation of a new DARPA research program focused on novel knowledge-discovery and data-mining (KDD) methods appropriate for counter-terrorism.

The scope of the new program was subsequently expanded to focus on three related sub-tasks in detecting potential terrorist activity from numerous large information sources in multiple formats. *Evidence extraction* (EE) is the task of obtaining structured evidence data from unstructured, natural-language documents. EE builds on information extraction technology developed under DARPA's earlier MUC (Message Understanding Conference) programs [Lehnert & Sundheim1991, Cowie & Lehnert1996] and the current ACE (Automated Content Extraction) program at the National Institute of Standards and Technology (NIST)[NIST]. *Link Discovery* (LD) is the task of identifying known, complex, multi-relational patterns that indicate potentially threatening activities in large amounts of relational data. Some of the input data for LD comes from EE, other input data comes from existing relational databases. Finally, *Pattern Learning* (PL) concerns the automated discovery of new relational patterns for potentially threatening activities. Novel patterns learned by PL can be used to improve the accuracy of LD. The current EELD program focused on these three sub-topics started in the summer of 2001. After 9/11, it was incorporated under the new Information Awareness Office (IAO) at DARPA.

The data and patterns used in EELD include representations of people, organizations, objects, and actions and many types of relations between them. The data is perhaps best represented as a large graph of entities connected by a variety of relations. The areas of *link analysis* and *social network analysis* in sociology, criminology, and intelligence [Jensen & Goldberg1998, Wasserman & Faust1994, Sparrow1991] study such networks using graph-theoretic representations. Data mining and pattern learning for counter terrorism therefore requires handling such multi-relational, graph-theoretic data.

Unfortunately, most current data-mining methods assume the data is from a single relational table and consists of flat tuples of items, as in market-basket analysis. This type of data is easily handled by machine learning techniques that assume a "propositional" (a.k.a "feature vector" or "attribute value") representation of examples [Witten & Frank1999]. *Relational data mining* (RDM) [Džeroski & Lavrač2001b], on the other hand, concerns mining data from multiple relational tables that are richly connected. Given the style of data needed for link discovery, pattern learning for link discovery requires *relational* data mining. The most widely studied methods for inducing relational patterns are those in *inductive logic programming* (ILP) [Muggleton1992, Lavrac & Dzeroski1994]. ILP concerns the induction of Horn-clause rules in first-order logic (i.e., logic programs) from data in first-order logic. This paper discusses our on-going work on applying ILP to link discovery as a part of the EELD project.

2 Inductive Logic Programming (ILP)

ILP is the study of learning methods for data and rules that are represented in first-order predicate logic. Predicate logic allows for quantified variables and relations and can represent concepts that are not expressible using examples described as feature vectors. A relational database can be easily translated into first-order logic and be used as a source of data for ILP [Wrobel2001]. As an example, consider the following rules, written in Prolog syntax (where the conclusion appears first), that define the uncle relation:

```
uncle(X,Y) :- brother(X,Z),parent(Z,Y).
uncle(X,Y) :- husband(X,Z),sister(Z,W),parent(W,Y).
```

The goal of *inductive logic programming* (ILP) is to infer rules of this sort given a database of background facts and logical definitions of other relations [Muggleton1992, Lavrac & Dzeroski1994]. For example, an ILP system can learn the above rules for uncle (the *target predicate*) given a set of positive and negative examples of uncle relationships and a set of facts for the relations parent, brother, sister, and husband (the *background predicates*) for the members of a given extended family, such as:

```
uncle(tom,frank), uncle(bob,john),
not uncle(tom,cindy), not uncle(bob,tom)
parent(bob,frank), parent(cindy,frank),
parent(alice,john), parent(tom,john),
brother(tom,cindy), sister(cindy,tom),
husband(tom,alice), husband(bob,cindy).
```

Alternatively, rules that logically define the brother and sister relations could be supplied and these relationships inferred from a more complete set of facts about only the "basic" predicates: `parent`, `spouse`, and `gender`.

If-then rules in first-order logic are formally referred to as *Horn clauses*. A more formal definition of the ILP problem follows:

- **Given:**
  - Background knowledge, B, a set of Horn clauses.
  - Positive examples, P, a set of Horn clauses (typically ground literals).
  - Negative examples, N, a set of Horn clauses (typically ground literals).
- **Find:** A hypothesis, H, a set of Horn clauses such that:
  - $\forall p \in P : H \cup B \models p$ (completeness)
  - $\forall n \in N : H \cup B \not\models n$ (consistency)

A variety of algorithms for the ILP problem have been developed [Džeroski & Lavrač2001a] and applied to a variety of important data-mining problems [Džeroski2001]. Nevertheless, relational data mining remains an under-appreciated topic in the larger KDD community. For example, recent textbooks on data mining [Han & Kamber2001, Witten & Frank1999, Hand, Mannila, & Smyth2001] hardly mention the topic. Therefore, we believe it is an important topic for "next generation" data mining systems. In particular, it is critical for link discovery applications in counter-terrorism.

3 Initial Work on ILP for Link Discovery

We tested different ILP algorithms on various EELD datasets. The current EELD datasets pertain to two domains – Nuclear Smuggling and Contract Killing. The Contract-Killing domain is further divided into natural (real world) data manually collected and extracted from news sources and synthetic (artificial) data generated by simulators. Section 3.1 presents our experimental results on the natural Smuggling and Contract-Killing data, while section 3.2 presents our initial results on the synthetic Contract-Killing data.

3.1 Experiments on Natural Data

3.1.1 The Nuclear-Smuggling Data

The Nuclear-Smuggling dataset consists of reports on Russian nuclear materials smuggling [McKay, Woessner, & Roule2001]. The Chronology of Nuclear and Radioactive Smuggling Incidents is the basis for the analysis of patterns in the smuggling of Russian nuclear materials. The information in the Chronology is based on open-source reporting, primarily World News Connection (WNC) and Lexis-Nexis. There are also some articles obtained from various sources that have been translated from Italian, German and Russian. The research from which the Chronology grew began in 1994 and the chronology itself first appeared as an appendix to a paper by Williams and Woessner in 1995 [Williams & Woessner1995b, Williams & Woessner1995a]. The continually evolving Chronology then was published twice as separate papers in the same journal as part of the "Recent Events" section [Woessner1995, Woessner1997]. As part of the Evidence Extraction and Link Discovery (EELD) project, the coverage of the Chronology was extended to March 2000 and the Chronology itself grew to 572 incidents. The incident descriptions in the Chronology are one entry descriptions per incident. The incidents in the Chronology have also been extensively cross-referenced.

The data is presented as a chronology of the incidents in a relational database format. This format contains Objects (described in rows in tables), each of which has Attributes of differing types (i.e., columns in the tables), the values of which are a matter of input from the source information or from the user. The Objects are of different types, which are denoted by prefixes (E_, EV_, LK_, and L_), and consist of the following.

- Entity Objects (E_...): these consist of E_LOCATION, E_MATERIAL, E_ORGANIZATION, E_PERSON, E_SOURCE, and E_WEAPON;
- Event Objects (EV_...): these currently consist of the generic EV_EVENT;
- Link Objects (LK_...): used for expressing links between/among Entities and Events, and currently consisting of those represented by X's in Table 3.1.1.

Table 1: Links among Entities and Events in Nuclear-Smuggling Data

	Event	Person	Organization	Location	Weapon	Material
Event	X					
Person	X	X				
Organization	X	X	X			
Location	X	X	X	X		
Weapon	X	X	X	X	X	
Material	X	X	X	X	X	X

The actual database we use in our experiments has over 40 relational tables. The number of tuples in a relational table vary from 800 to as little as 2 or 3 elements.

The ILP system has to learn which events in an incident are *related* in order to construct larger knowledge structures that can be recognized as threats. Hence the ILP system needs positive training examples that specify "links" between events. We assume all other events are unrelated and therefore compose a set of negative examples. We stipulate that *related* is commutative. Therefore we specified to the ILP system used in our experiments that `related(B,A)` is true if `related(A,B)` is proven, and vice-versa. Our set of examples consists of 140 positive examples and 140 distinct negative examples randomly drawn from a full set of 8124 negative examples.

The linking problem in the Nuclear-Smuggling data is thus quite challenging in that it is a heavily relational learning problem over a large number of relations, whereas traditional ILP applications usually require a small number of relations.

3.1.2 The Natural Contract-Killing Data

The dataset of contract killings was first compiled by O'Hayon and Cook [Cook & O'Hayon2000]. It was a response to research on Russian organized crime that encountered frequent and often tantalizing references to contract killings. Each of the contract-killing reports provided a still photograph of the criminal scene in Russia, but there was no comparable assessment of how these were linked, what the trends were, who the victims were, the relationship between victims themselves or the relationship between victims and perpetrators. The dataset on contract killings has been continually expanded by Cook and O'Hayon with funding from DARPA's EELD program through Veridian Systems Division (VSD) [Williams2002]. The database was captured as a "chronology" of the incidents. Each incident in the chronology received a description of the information drawn from the sources, typically one news article, but occasionally more than one. As in the Nuclear-Smuggling dataset, information in the chronology is based on open-source reporting, especially Foreign Broadcast Information Service (FBIS) and Joint Publications Research Service (JPRS) journals, and subsequently both FBIS on-line and the cut-down on-line version World News Connection (WNC). These services and Lexis-Nexis are the main information sources. Additional materials on the worldwide web were consulted when this was feasible and helpful. The search was as exhaustive as possible given the limited time and resources of those involved.

The data is organized in relational tables in the same format as the Nuclear-Smuggling data described in the previous section. The dataset used in our experiments has 48 relational tables. The number of tuples in a relational table varies from 1,000 to as little as 1 element. The ILP learner task was to characterize Rival versus Obstacle plus Threat events (i. e., the Obstacle and Threat examples were pooled into one category, thereby producing a two-category learning task). Rival, Obstacle, and Threat are treated as "motives" in the dataset. The motivation to this learning task thus is to recognize patterns of activity that indicate underlying motives, which in turn contributes to recognizing threats. The number of positive examples in this dataset is 38, while the number of negative examples is 34.

3.1.3 ILP Results on the Natural Data

Aleph We use the ILP system Aleph [Srinivasan2001] to learn rules to the natural datasets. By default, Aleph uses a simple greedy set covering procedure that constructs a complete and consistent hypothesis one clause at a time. In the search for any single clause, Aleph selects the first uncovered positive example as the seed example, "saturates" this example, and performs an admissible search over the space of clauses that subsume this saturation, subject to a user-specified clause length bound. Further details about our use of Aleph in these experiments are available in [Dutra, Page, & V. Santos Costa2002].

Ensembles *Ensembles* aim at improving accuracy through combining the predictions of multiple classifiers in order to obtain a single classifier. Therefore, we also investigate employing an ensemble of classifiers, where each classifier is a logical theory generated by Aleph. Many methods have been presented for ensemble generation [Dietterich1998]. In this paper, we concentrate on a popular method that is known to frequently create a more accurate ensemble than individual components, *bagging* [Breiman1996a]. Bagging works by training each classifier on a random sample from the training set. Bagging has the important advantage that it is effective on "unstable learning algorithms" [Breiman1996b], where small variations in parameters can cause huge variations in the learned theories. This is the case with ILP. A second advantage is that it can be implemented in parallel trivially. Further details about our bagging approach within ILP, as well as our experimental methodology, can be found in [Dutra, Page, & V. Santos Costa2002]. Our experimental results are based on a five-fold cross-validation, where five times we train on 80% of the examples and then test what was learned on the remaining 20% (in addition, each example is in one and only one test set).

For the task of identifying linked events within the Nuclear-Smuggling dataset, Aleph produces an average testset accuracy of 85%. This is an improvement over the baseline case (majority class—always guessing two events are not linked), which produces an average accuracy of 78%. Bagging (with 25 different sets of rules) increases the accuracy to almost 90%.

An example of a rule with good accuracy found by the system is shown in Figure 1. This rule covers 43 of the 140 positive examples and no negative examples. According to this rule, two smuggling events `A` and `E` are related if event `A` involves a person `C` who is also involved in another event `F`. Event `F` involves some material `G` that appears

```
linked(A,E) :-
   lk_event_person(_,EventA,PersonC,_,RelationB,RelationB,DescriptionD),
   lk_event_person(_,EventF,PersonC,_,RelationB,RelationB,DescriptionD),
   lk_material_location(_,MaterialG,_,EventE,_,_,_,_,_),
   lk_event_material(_,EventF,MaterialG,_,_,_,_).
```

Figure 1: Nuclear-Smuggling Data: Sample Learned Rule

in event E. In other words, a person C in event A is involved in a third event F that uses material from event E. Person C played the same role B, with description D, in events A and F. The "_" symbols mean that those arguments were not relevant for that rule. Figure 2 illustrates the connections between events, material and people involved. Solid lines are direct connections shown by the literals in the body of the clause. The dotted line corresponds to the new concept learned that describes connection between two events.

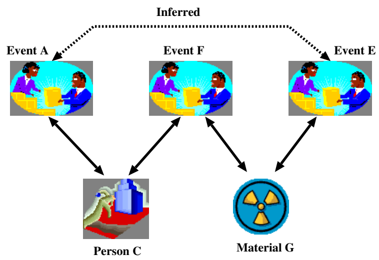

Figure 2: Pictorial representation of a learned rule.

The task of identifying motive in the Contract-Killing data set is much more difficult, with Aleph's accuracy at 59%, compared with the baseline accuracy of 52%. Again the utilization of ensembles improves the accuracy, this time to 69%. The rule in Figure 3 shows one kind of logical clause the ILP system we use found for this dataset.

The rule covers 19 of the 38 positive examples and a single negative example. The rule says that event A is a killing by a rival if we can follow a chain of events that connects event A to event B, event B to event E, and event E to an event F that relates two organizations. Events A and E have the same kind of relation, RelationC, to B. All events in the chain are subsets of the same incident D.

3.2 Experiments on Synthetic Data

The synthetic data for Contract Killing are provenient from two different simulators. One is a Bayesian Network (BN) Simulator [Extraction & Transport2002] and the second one is a Task-Based (TB) Simulator [Team, Lead, & Powers2002]. The Bayesian Network simulator generates data based on a probabilistic model developed by Information Extraction and Transport Incorporated (IET). The BN simulator outputs case files, which contain complete and unadulterated descriptions of each murder case.

```
rivalKilling(EventA) :-
     lk_event_event(_,EventB,EventA,RelationC,EventDescriptionD),
     lk_event_event(_,EventB,EventE,RelationC,EventDescriptionD),
     lk_event_event(_,EventE,EventF,_,EventDescriptionD),
     lk_org_org(_,_,_,EventF,_,_,_,_,_).
```

Figure 3: Natural Contract-Killing Data: Sample Learned Rule

These case files are then filtered for observability, so that facts that would not be accessible to an investigator are eliminated. To make the task more realistic this data is also corrupted, e.g., by misidentifying role players or incorrectly reporting group memberships. This filtered and corrupted data form the evidence files. In our representation of the evidence files, facts about each event are represented as binary predicates, such as:

```
isa(murder714, murder_for_hire)
perpetrator(murder714, killer186)
victim(murder714, murder_victim996)
deviceTypeUsed(murder714, pistol_czech)
```

The Task-Based simulator [Team, Lead, & Powers2002] provides a flexible mechanism for creating synthetic datasets within the EELD program. It includes a pattern specification language, a Knowledge Base, case generation and representation, evidence generation and corruption, and answer key representations. Whereas the Bayesian simulator relied on a Bayesian network, the core of the task-based simulator are *tasks*. Each task contains one or more methods, where each method has a probability of being selected given that its preconditions are satisfied. The simulator also provides powerful functionality for filtering and corrupting data. This is particularly important to represent situations where actual data is expected to have low observability.

As shown next, we also represent simulation output as binary predicates:

```
report_on_situation(uid6147).
starting_date(uid6147,"1/15/2002").
information_source_type(uid6147,police_organization).
meeting_taking_place(uid6146).
date_of_event(uid6146,"1/9/2002").
social_participants(uid6146,uid4126).
social_participants(uid6146,uid3152).
ite_illocutionary_force(uid6146,inform).
```

Notice that the simulation results include meta-data, such as when and where a specific event was reported. We do not take advantage of that data in our current experiments.

3.2.1 ILP Results on the Synthetic BN-based Data

The synthetic BN-based contract killing dataset that we used consists of 200 murder events. Each murder event has been labeled as a murder for hire, first-degree or second-degree murder. There are 71 murder for hire events, 75 first-degree and 54 second-degree murder events. Our task was to learn a classifier to correctly classify an unlabeled event into one of these three categories.

For this task, we used a variation of mFoil [Lavrac & Dzeroski1994] to learn a binary classifier to discriminate between events that are murder for hire and events that are not. Like Aleph, mFoil learns one clause at a time using greedy covering, but uses a constrained, general-to-specific search to learn individual rules. We also used mFoil to learn two more classifiers to identify first-degree and second-degree murders. The three binary classifiers are combined to form a three-way classifier for the task. If an event is classified as a positive example by only one classifier then the event is labeled with the category corresponding to that classifier. If more than one classifier classifies an event as a positive example then we select the category more commonly represented in the training data.

We ran 10-fold cross-validation on the dataset of 200 murder events. We measured the precision and recall of our classifier for each of the three categories. Precision and recall for a category is defined below:

$$Precision_C = \frac{number\ of\ events\ correctly\ classified\ as\ C}{number\ of\ events\ classified\ as\ C} \times 100\%$$

$$Recall_C = \frac{number\ of\ events\ correctly\ classified\ as\ C}{number\ of\ C\ events} \times 100\%$$

The results are summarized in Table 2. We observe that apart from recall for second-degree murders, the precision and recall results are all above 85%. Our system learns a very precise classifier for second-degree murders, but as a consequence it has a lower recall. However, we can adjust the parameters of our system to compromise precision for higher recall.

We also computed the accuracy of our classifier, which is defined as the percentage of events correctly classified into one of the three categories. We compare this to the majority-class classifier, which always classifies events as the most frequently represented category. In our experiments the accuracy of the majority-class classifier is 38%. And the classification accuracy of our system is 77% which is more than twice that of the majority-class classifier.

Figure 4 shows some of the sample rules that our system learns. According to the first rule a murder event that involves a member of a criminal organization and that is associated with another crime that was motivated by economic gains is a murder for hire. The second rule says that if a murder is the result of an event that was performed by someone in love, then it is a first-degree murder (as these are mainly premeditated murders). According to the third rule if a murder is the result of a theft that is motivated by rivalry and that is performed on public property then it is a second-degree murder. These sample rules show that not only does our system do well in classifying the different events, it also produces rules that are meaningful and interpretable by humans.

3.2.2 ILP Results on the Synthetic TB Data

The synthetic TB contract killing dataset that we used consists of several runs of the task-based simulator. Parameters were set to very low observability and to large simu-

Table 2: Results on the Synthetic Contract-Killing Data

	Murder for hire	**1st degree**	**2nd degree**
Precision	86%	91%	96%
Recall	91%	88%	59%

```
murder_for_hire(A) :-
  group_member_maleficiary(A, B),
  sub_events(A, C),
  crime_motive(C, economic).

first_degree_murder(A) :-
  sub_events(A, B),
  performed_by(B, C),
  loves(C, D).

second_degree_murder(A) :-
  sub_events(A, B),
  event_occurs_at_location_type(B, publicProperty),
  crime_motive(B, rival),
  occurrent_subevent_type(B, stealing_Generic).
```

Figure 4: Synthetic Contract-Killing Data: Sample Learned Rules

lations. We were interested in learning how to detect instances of murder for hire. We define each instance to contain a victim, a perpetrator, and a contractor. Positive examples were obtained from the simulator's output. Because it is rather hard to define what were the interesting cases of non-murder for hires, we did not use negative examples for our initial experiments. Instead, we performed positive-only learning.

The simulator's output includes a large number of relations, about 200 relations. Each simulation generates background knowledge consisting of around 30,000 facts. The number of positive examples also varies, ranging between 15 and 20 per simulation.

Experiments were performed with Aleph using 10-fold cross-validation. We used Aleph's implementation of Muggleton's positive data learning algorithm [Muggleton2001], since there were no explicit negative examples. As mentioned before, each fold corresponds to an independent run of the simulator. All simulator generated constants were renamed uniquely across folds to avoid duplicate names in different folds.

Figure 5 shows a sample rule that the Aleph system learned (notice that the rules were obtained with very low observability). The rule detects a murder for hire if the Perpetrator is known to have commited a crime, and if the Contractor met with someone whose bank account was used both to receive and transfer money. This rule covers 97 out of 165 positive examples.

The average accuracy over the 10 folds was 80%. Most of the best rules Aleph generated focused on money transactions involving the perpetrator or the contractor.

```
murder_for_hire(VictimA,ContractorB,PerpetratorC) :-
   perpetrator(CrimeD,PerpetratorC),
   social_participants(MeetingE,ContractorB),
   social_participants(MeetingE,PersonF),
   account_holder(AccountG,PersonF),
   from_generic(MoneyTransferH,AccountG),
   to_generic(MoneyTransferI,AccountG).
```

Figure 5: Task-Based Simulator Data: Sample Learned Rule

4 Current and Future Research

An under-studied issue in relational data mining is scaling algorithms to very large databases. Most research on ILP and RDM has been conducted in the machine learning and artificial intelligence (AI) communities rather than in the database and systems communities. Consequently, there has been insufficient research on systems issues involved in performing RDM in commercial relational-database systems and scaling algorithms to extremely large datasets that will not fit in main memory. Integrating ideas from systems work in data mining and deductive databases [Ramamohanarao & Harland1994] would seem to be critical in addressing these issues.

Related to scaling, we are currently working on efficiently learning complex relational concepts from large amounts of data by using stochastic sampling methods. A major shortcoming of ILP is the computational demand that results from the large hypothesis spaces searched. Intelligently sampling these large spaces can provide excellent performance in much less time [Srinivasan1999, Zelezny, Srinivasan, & Page2002].

We are also developing algorithms that learn more robust, probabilistic relational concepts represented as stochastic logic programs [Muggleton2002] and variants. This will enrich the expressiveness and robustness of learned concepts. As an alternative to stochastic logic programs, we are working on learning clauses in a constraint logic programming language where the constraints are Bayesian networks [Page2000, Costa, Page, & Cussens2002].

One approach that we plan to further investigate is the use of approximate prior knowledge to induce more accurate, comprehensible relational concepts from fewer training examples [Richards & Mooney1995]. The use of prior knowledge can greatly reduce the burden on users; they can express the "easy" aspects of the task at hand and then collect a small number of training examples to refine and extend this prior knowledge.

Finally, we plan to use active learning to allow our ILP systems to select more effective training examples for interactively learning relational concepts [Muggleton *et al.*1999]. By intelligently choosing the examples for users to label, better extraction accuracy can be obtained from fewer examples, thereby greatly reducing the burden on the users of our ILP systems.

5 Related Work

Although it is the most widely studied, ILP is not the only approach to relational data mining. In particular, other participants in the EELD program are taking alternative RDM approaches to pattern learning for link discovery. This section briefly reviews these other approaches.

5.1 Graph-based Relational Learning

Some relational data mining methods are based on learning structural patterns in graphs. In particular, SUBDUE [Cook & Holder1994, Cook & Holder2000] discovers highly repetitive subgraphs in a labeled graph using the minimum description length (MDL) principle. SUBDUE can be used to discover interesting substructures in graphical data as well as to classify and cluster graphs. Discovered patterns do not have to match the data exactly since SUBDUE can employ an inexact graph-matching procedure based on graph edit-distance. SUBDUE has been successfully applied to a number of important RDM problems in molecular biology, geology, and program analysis. It is also currently being applied to discover patterns for link discovery as a part of the EELD project (more details at `http://ailab.uta.edu/eeld/`). Since relational data for LD is easily represented as labeled graphs, graph-based RDM methods like SUBDUE are a natural approach.

5.2 Probabilistic Relational Models

Probabilistic relational models (PRM's) [Koller & Pfeffer1998] are an extension of Bayesian networks for handling relational data. Methods for learning Bayesian networks have also been extended to produce algorithms for inducing PRM's from data [Friedman *et al.*1999]. PRM's have the nice property of integrating some of the advantages of both logical and probabilistic approaches to knowledge representation and reasoning. They combine some of the representational expressivity of first-order logic with the uncertain reasoning abilities of Bayesian networks. PRM's have been applied to a number of interesting problems in molecular biology, web-page classification, and analysis of movie data. They are also currently being applied to pattern learning for link discovery as a part of the EELD project.

5.3 Relational Feature Construction

One approach to learning from relational data is to first "flatten" or "propositionalize" the data by constructing features that capture some of the relational information and then applying a standard learning algorithm to the resulting feature vectors [Kramer, Lavrač, & Flach2001]. PROXIMITY [Neville & Jensen2000] is a system that constructs features for categorizing entities based on the categories and other properties of other entities to which it is related. It then uses an interactive classification procedure to dynamically update inferences about objects based on earlier inferences about related objects. PROXIMITY has been successfully applied to company and movie data. It is also currently being applied to pattern learning for link discovery as a part of the EELD project.

6 Conclusions

Link discovery is an important problem in automatically detecting potential threatening activity from large, heterogeneous data sources. The DARPA EELD program is a U.S. government research project exploring link discovery as an important problem in the development of new counter-terrorism technology. Learning new link-discovery patterns that indicate potentially threatening activity is a difficult data mining problem. It requires discovering novel relational patterns in large amounts of complex relational data. Most existing data-mining methods assume flat data from a single relational table and are not appropriate for link discovery. Relational data mining techniques, such as inductive logic programming, are needed. Many other problems in molecular biology [Srinivasan *et al.*1996], natural-language understanding [Zelle & Mooney1996], web page classification [Craven *et al.*2000], information extraction [Califf & Mooney1999, Freitag1998], and other areas also require mining multi-relational data. However, relational data mining requires exploring a much larger space of possible patterns and performing complex inference and pattern matching. Consequently, current RDM methods are not scalable to very large databases. Consequently, we believe that relational data mining is one of the major research topics in the development of the next generation of data mining systems, particularly those in the area of counter-terrorism.

Acknowledgments

This research is sponsored by the Defense Advanced Research Projects Agency and managed by Rome Laboratory under contract F30602-01-2-0571. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency, Rome Laboratory, or the United States Government.

Vítor Santos Costa and Inês de Castro Dutra are on leave from COPPE/Sistemas, Federal University of Rio de Janeiro and were partially supported by CNPq. Many thanks to Hans Chalupksy's group at ISI, in particular to André Valente, who gave us support on using the Task-based simulator. We would like to thank the Biomedical Computing Group support staff and the Condor Team at the Computer Sciences Department of the University of Wisconsin, Madison, for their invaluable help with Condor. We also would like to thank Ashwin Srinivasan for his help with the Aleph system.

References

[Breiman1996a] Breiman, L. 1996a. Bagging Predictors. *Machine Learning* 24(2):123–140.

[Breiman1996b] Breiman, L. 1996b. Stacked Regressions. *Machine Learning* 24(1):49–64.

[Califf & Mooney1999] Califf, M. E., and Mooney, R. J. 1999. Relational learning of pattern-match rules for information extraction. In *Proceedings of the 17th National Conference on Artificial Intelligence*, 328–334.

[Cook & Holder1994] Cook, D. J., and Holder, L. B. 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research* 1:231–255.

[Cook & Holder2000] Cook, D. J., and Holder, L. B. 2000. Graph-based data mining. *IEEE Intelligent Systems* 15(2):32–41.

[Cook & O'Hayon2000] Cook, W., and O'Hayon, G. 2000. Chronology of Russian killings. *Transnational Organized Crime* 4(2).

[Costa, Page, & Cussens2002] Costa, V. S.; Page, D.; and Cussens, J. 2002. CLP(BN): Constraint logic programming with Bayesian network constraints. Unpublished Technical Note.

[Cowie & Lehnert1996] Cowie, J., and Lehnert, W. 1996. Information extraction. *Communications of the ACM* 39(1):80–91.

[Craven *et al.*2000] Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A. K.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence* 118(1-2):69–113.

[Dietterich1998] Dietterich, T. G. 1998. Machine-learning research: Four current directions. *The AI Magazine* 18(4):97–136.

[Dutra, Page, & V. Santos Costa2002] Dutra, I. C.; Page, D.; and V. Santos Costa, J. S. 2002. An empirical evaluation of bagging in inductive logic programming. In *Proceedings of the 12th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence. Springer-Verlag.

[Džeroski & Lavrač2001a] Džeroski, S., and Lavrač, N. 2001a. An introduction to inductive logic programming. In Džeroski, S., and Lavrač, N., eds., *Relational Data Mining*. Berlin: Springer Verlag.

[Džeroski & Lavrač2001b] Džeroski, S., and Lavrač, N., eds. 2001b. *Relational Data Mining*. Berlin: Springer Verlag.

[Džeroski2001] Džeroski, S. 2001. Relational data mining applications: An overview. In Džeroski, S., and Lavrač, N., eds., *Relational Data Mining*. Berlin: Springer Verlag.

[Extraction & Transport2002] Extraction, I., and Transport, I. 2002. Bayesian netwrok simulator. Internal report.

[Freitag1998] Freitag, D. 1998. Information extraction from HTML: Application of a general learning approach. In *Proceedings of the 16th National Conference on Artificial Intelligence*, 517–523. Madison, WI: AAAI Press / The MIT Press.

[Friedman *et al.*1999] Friedman, N.; Getoor, L.; Koller, D.; and Pfeffer, A. 1999. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.

[Han & Kamber2001] Han, J., and Kamber, M. 2001. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kauffmann Publishers.

[Hand, Mannila, & Smyth2001] Hand, D. J.; Mannila, H.; and Smyth, P. 2001. *Principles of Data Mining*. Cambridge, MA: MIT Press.

[Jensen & Goldberg1998] Jensen, D., and Goldberg, H., eds. 1998. *AAAI Fall Symposium on Artificial Intelligence for Link Analysis*. Menlo Park, CA: AAAI Press.

[Koller & Pfeffer1998] Koller, D., and Pfeffer, A. 1998. Probabilistic frame-based systems. In *Proceedings of the 16th National Conference on Artificial Intelligence*, 580–587. Madison, WI: AAAI Press / The MIT Press.

[Kramer, Lavrač, & Flach2001] Kramer, S.; Lavrač, N.; and Flach, P. 2001. Propositionalization approaches to relational data mining. In Džeroski, S., and Lavrač, N., eds., *Relational Data Mining*. Berlin: Springer Verlag.

[Lavrac & Dzeroski1994] Lavrac, N., and Dzeroski, S. 1994. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

[Lehnert & Sundheim1991] Lehnert, W., and Sundheim, B. 1991. A performance evaluation of text-analysis technologies. *AI Magazine* 12(3):81–94.

[McKay, Woessner, & Roule2001] McKay, S. J.; Woessner, P. N.; and Roule, T. J. 2001. Evidence extraction and link discovery (EELD) seedling project, database schema description, version 1.0. Technical Report 2862, Veridian Systems Division.

[Muggleton *et al.*1999] Muggleton, S.; Bryant, C.; Page, C.; and Sternberg, M. 1999. Combining active learning with inductive logic programming to close the loop in machine learning. In Colton, S., ed., *Proceedings of the AISB'99 Symposium on AI and Scientific Creativity (informal proceedings)*.

[Muggleton1992] Muggleton, S. H., ed. 1992. *Inductive Logic Programming*. New York, NY: Academic Press.

[Muggleton2001] Muggleton, S. 2001. Learning From Positive Data. *Machine Learning Journal*. Accepted for Publication subject to revision.

[Muggleton2002] Muggleton, S. 2002. Stochastic logic programs. *Journal of Logic Programming*. To appear.

[Neville & Jensen2000] Neville, J., and Jensen, D. 2000. Iterative classification in relational data. In *Papers from the AAAI-00 Workshop on Learning Statistical Models from Relational Data*. Austin, TX: AAAI Press / The MIT Press.

[NIST] NIST. ACE - Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/.

[Page2000] Page, D. 2000. ILP: Just do it! In Lloyd, J.; Dahl, V.; Furbach, U.; Kerber, M.; Lau, K.-K.; Palamidessi, C.; Pereira, L.; Sagiv, Y.; and Stuckey, P., eds., *Proceedings of Computational Logic 2000*, 25–40. Springer Verlag.

[Ramamohanarao & Harland1994] Ramamohanarao, K., and Harland, J. 1994. An introduction to deductive database languages and systems. *VLDB Journal* 3:2.

[Richards & Mooney1995] Richards, B. L., and Mooney, R. J. 1995. Automated refinement of first-order Horn-clause domain theories. *Machine Learning* 19(2):95–131.

[Sparrow1991] Sparrow, M. K. 1991. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks* 13:251–274.

[Srinivasan *et al.*1996] Srinivasan, A.; Muggleton, S. H.; Sternberg, M. J.; and King, R. D. 1996. Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence* 85:277–300.

[Srinivasan1999] Srinivasan, A. 1999. A study of two sampling methods for analysing large datasets with ILP. *Data Mining and Knowledge Discovery* 3(1):95–123.

[Srinivasan2001] Srinivasan, A. 2001. *The Aleph Manual*.

[Team, Lead, & Powers2002] Team, E. P. E.; Lead, I. P. E.; and Powers, J. 2002. Task-based simulator version 9.0, 15 july 2002. Internal report, Information Extraction and Transport, Inc.

[Wasserman & Faust1994] Wasserman, S., and Faust, K. 1994. *Social Network Analysis: Methods & Applications*. Cambridge, UK: Cambridge University Press.

[Williams & Woessner1995a] Williams, P., and Woessner, P. N. 1995a. Nuclear material trafficking: An interim assessment. *Transnational Organized Crime* 1(2):206–238.

[Williams & Woessner1995b] Williams, P., and Woessner, P. N. 1995b. Nuclear material trafficking: An interim assessment, ridgway viewpoints. Technical Report 3, Ridgway Center, University of Pittsburgh.

[Williams2002] Williams, P. 2002. Patterns, indicators, and warnings in link analysis: The contract killings dataset. Technical Report 2878, Veridian Systems Division.

[Witten & Frank1999] Witten, I. H., and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann.

[Woessner1995] Woessner, P. N. 1995. Chronology of nuclear smuggling incidents: July 1991-may 1995. *Transnational Organized Crime* 1(2):288–329.

[Woessner1997] Woessner, P. N. 1997. Chronology of radioactive and nuclear materials smuggling incidents: July 1991-june 1997. *Transnational Organized Crime* 3(1):114–209.

[Wrobel2001] Wrobel, S. 2001. Inductive logic programming for knowledge discovery in databases. In Džeroski, S., and Lavrač, N., eds., *Relational Data Mining*. Berlin: Springer Verlag.

[Zelezny, Srinivasan, & Page2002] Zelezny, F.; Srinivasan, A.; and Page, D. 2002. Lattice-search runtime distributions may be heavy-tailed. In *Proceedings of the 12th International Conference on Inductive Logic Programming*. Springer Verlag.

[Zelle & Mooney1996] Zelle, J. M., and Mooney, R. J. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the 14th National Conference on Artificial Intelligence*, 1050–1055.