

Qualitative Comparison of Graph-based and Logic-based Multi-Relational Data Mining: A Case Study

Nikhil S. Ketkar
University of Texas at Arlington
ketkar@cse.uta.edu

Lawrence B. Holder
University of Texas at Arlington
holder@cse.uta.edu

Diane J. Cook
University of Texas at Arlington
cook@cse.uta.edu

ABSTRACT

The goal of this paper is to generate insights about the differences between graph-based and logic-based approaches to multi-relational data mining by performing a case study of graph-based system, Subdue and the inductive logic programming system, CProgol. We identify three key factors for comparing graph-based and logic-based multi-relational data mining; namely, the ability to discover structurally large concepts, the ability to discover semantically complicated concepts and the ability to effectively utilize background knowledge. We perform an experimental comparison of Subdue and CProgol on the Mutagenesis domain and various artificially generated Bongard problems. Experimental results indicate that Subdue can significantly outperform CProgol while discovering structurally large multi-relational concepts. It is also observed that CProgol is better at learning semantically complicated concepts and it tends to use background knowledge more effectively than Subdue.

1. INTRODUCTION

Multi-relational data mining (MRDM)[4] is a subfield of data mining which focuses on knowledge discovery from relational databases comprising multiple tables. Representation is a fundamental as well as a critical aspect in the process of discovery and two forms of representation, namely the graph-based representation and the logic-based representation, have been used for MRDM. Logic-based MRDM popularly known as Inductive Logic Programming (ILP) [8] is characterized by the use of logic for the representation of multi-relational data. ILP systems represent examples, background knowledge, hypotheses and target concepts in Horn clause logic. The core of ILP is the use of logic for representation and the search for syntactically legal hypotheses constructed from predicates provided by the background knowledge. ILP systems such as FOIL [12], CProgol [9], Golem[11] and WARMR[3] have been extensively applied to supervised learning and to a certain extent to unsupervised learning.

Graph-based approaches are characterized by representation of multi-relational data in the form of graphs. Graph-based approaches represent examples, background knowledge, hypotheses and target concepts as graphs. The core of graph-based approaches is the use of a graph-based representation and the search for graph patterns which are frequent or which compress the input graphs or which distinguish positive and negative examples. Graph-based MRDM systems such as Subdue[2], FSG[6], gSpan[16], GBI[7], and AGM[5] have been extensively applied to unsupervised learning and to a certain extent to supervised learning.

The goal of the paper is to perform a qualitative comparison of graph-based and logic-based MRDM, supported by extensive experimentation. The paper identifies the specific qualitative dimensions on which two major paradigms of multi-relational data mining differ. CProgol is selected as a representative of logic-based approaches and Subdue is selected as a representative of graph-based approaches. Experiments are performed on the Mutagenesis dataset which is a benchmark dataset for MRDM. In most of the experiments, transformations are applied to the Mutagenesis dataset or distinct types of background knowledge are provided to Subdue and CProgol. The rationale behind doing so is to perform lesion studies and gain insight on the specific abilities of the approaches. Additional experiments are performed on artificially generated Bongard problems to reinforce the findings from the experiments on the Mutagenesis dataset. We analyze the experimental data to generate insights about the fundamental differences between the approaches underlying their strengths and weaknesses.

The rest of the paper is organized as follows. In Section 2, we identify the factors on the basis of which the graph-based and logic-based approaches should be compared, namely, the ability to learn structurally large concepts, the ability to learn semantically complicated concepts and the ability to effectively utilize background knowledge. Section 3 describes the experimental setup, the MRDM systems Subdue and CProgol, the Mutagenesis Dataset and the Bongard problems. Sections 4, 5 and 6 describe the experiments, present the results, and analyze the approaches based on the three comparison factors. Conclusions and future work are presented in Section 7.

2. FACTORS FOR COMPARISON

By performing a comparison of the graph-based and logic-based approaches to MRDM, we intended to analyze the ability of the approaches to efficiently discover complex multi-relational concepts and to effectively utilize back-

This paper appears in the Proceedings of the Fourth International Workshop on Multi-Relational Data Mining (MRDM-2005), August 21, 2005, Chicago. The proceedings were edited by Sašo Džeroski and Hendrik Blockeel. The paper is published here with the permission of the authors, who retain the copyright of this material.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2005		2. REPORT TYPE		3. DATES COVERED 00-00-2005 to 00-00-2005	
4. TITLE AND SUBTITLE Qualitative Comparison of Graph-based and Logic-based Multi-Relational Data Mining: A Case Study				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The University of Texas at Arlington, Department of Computer Science and Engineering, Box 19015, Arlington, TX, 76019				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

ground knowledge. For doing so it is essential to establish some notions on the complexity of a multi-relational concept and to identify the types of background knowledge generally available in the task of MRDM.

The complexity of a multi-relational concept is a direct consequence of the number of relations in the concept. A multi-relational concept is more complicated to learn than some other multi-relational concept if learning that concept involves learning more relations than the other concept. For example learning the concept of arene (six member ring as in benzene) which comprises learning six relations, involves the exploration of a larger hypothesis space than learning the concept of hydroxyl (oxygen connected to hydrogen as in methanol), which comprises learning one relation. The concept of arene is thus more complicated than that of hydroxyl. Although the number of relations in the multi-relational concept is a key factor in the complexity of the multi-relational concept, there are also other factors such as the number of relations in the examples from which the concept is to be learned. For example, learning the concept of hydroxyl from a set of large molecules (e.g., phenols, etc.) involves the exploration of a larger hypothesis space than learning the same hydroxyl concept from a set of small molecules (e.g., methanol, etc.). The concept of hydroxyl group is thus more complicated to learn from phenols than it is from a set of alcohols. We identify this complexity as structural complexity.

In order to learn a particular concept, it is essential that the representation used by a multi-relational data mining system is able to express that particular concept. For a representation to express a particular concept, it is beneficial to have both the syntax which expresses the concept and the semantics which associates meaning to the syntax. The concepts which cannot be represented by the representation used by the MRDM system can be explicitly instantiated in the examples. A relational concept can be said to have a higher complexity than some other relational concept if representing that concept requires a more expressive representation. For example to learn numerical ranges, it is essential to have the syntax and the semantics for representing notions like 'lesser than', 'greater than' and 'equal to'. We identify this complexity as semantic complexity.

A relational learner can be provided background knowledge which condenses the hypothesis space. For example if the concept to be learned is 'compounds with three arene rings' (six member ring as in benzene) and the concept of an arene ring is provided as a part of the background knowledge, then the arene rings in examples could be condensed to a single entity. This would cause a massive reduction in the hypothesis space required to be explored to learn the concept and the relational learner would perform more efficiently than without the background knowledge. We identify such background knowledge as background knowledge intended to condense the hypothesis space.

A relational learner can be provided background knowledge which augments the hypothesis space. For example consider that the relational learner is provided with background knowledge which allows it to learn concepts like 'lesser than', 'greater than' and 'equal to'. In this case, the relational learner would explore a hypothesis space larger than what it would explore without the background knowledge. Thus introducing background knowledge has augmented the hypothesis space and has facilitated the learning

of concepts which would not be learned without the background knowledge. We identify such background knowledge as background knowledge intended to augment the hypothesis space.

Using these notions, we now identify the factors on the basis of which the graph-based approach and the logic-based approach can be compared. They are,

1. Ability to learn structurally large relational concepts.
2. Ability to learn semantically complicated relational concepts or the ability to effectively use background knowledge that augments the hypothesis space to learn semantically complicated relational concepts.
3. Ability to effectively use background knowledge that condenses the hypothesis space.

3. EXPERIMENTAL SETUP

In this section we briefly discuss CProlog, Subdue, the Mutagenesis domain and the Bongard problems which are used for the experimental comparison.

3.1 CProlog

CProlog[9] is an ILP system, characterized by the use of mode-directed inverse entailment and a hybrid search mechanism. Inverse entailment is a procedure which generates a single, most specific clause that, together with the background knowledge, entails the observed data. The inverse entailment in CProlog is mode-directed that is, it uses mode definitions. A mode declaration is a constraint which imposes restrictions on the atoms and their arguments appearing in a hypothesis clause by,

1. Determining which atoms can occur in the head and the body of hypotheses.
2. Determining which arguments can be input variables, output variables or constants.
3. Determining the number of alternative solutions for instantiating the atom.

The user-defined mode declarations aid the generation of the most specific clause. CProlog first computes the most specific clause which covers the seed example and belongs to the hypothesis language. The most specific clause can be used to bound the search from below. The search is now bounded between the empty clause and the most specific clause. The search proceeds within the bounded θ -subsumption lattice in a general-to-specific manner. The search is a hybrid search, because it is a general-to-specific search bounded from below with respect to the most specific clause. The search strategy is an A* algorithm which is guided by a weighted compression and accuracy measure. The A* search returns a clause which covers the most positive examples and maximally compresses the data. Any arbitrary Prolog program can serve as background knowledge for CProlog.

3.2 Subdue

Subdue[2] is a graph-based MRDM system capable of unsupervised and supervised learning. When operating as a supervised learner, Subdue finds substructures distinguishing the positive and negative examples. Subdue performs a beam search which begins from substructures consisting

of all vertices with unique labels. The substructures are extended by one vertex and one edge or one edge in all possible ways, as guided by the input graph, to generate candidate substructures. Subdue maintains the instances of substructures (in order to avoid subgraph isomorphism) and uses graph isomorphism to determine the instances of the candidate substructure in the input graph. Candidate substructures are evaluated according to classification accuracy or the minimum description length principle [14]. The length of the search beam determines the number of candidate substructures retained for further expansion. This procedure repeats until all substructures are considered or the user imposed computational constraints are exceeded. At the end of this procedure the positive examples covered by the best substructure are removed. The process of finding substructures and removing positive examples continues until all the positive examples are covered.

3.3 Mutagenesis Domain

The Mutagenesis dataset[15] has been collected to identify mutagenic activity in a compound based on its molecular structure and is considered to be a benchmark dataset for MRDM. The Mutagenesis dataset consists of the molecular structure of 230 compounds, of which 138 are labeled as mutagenic and 92 as non-mutagenic. The mutagenicity of the compounds has been determined by the Ames Test. The task is to distinguish mutagenic compounds from non-mutagenic ones based on their molecular structure. The Mutagenesis dataset basically consists of atoms, bonds, atom types, bond types and partial charges on atoms. The dataset also consists of the hydrophobicity of the compound (logP), the energy level of the compound’s lowest unoccupied molecular orbital (LUMO), a boolean attribute identifying compounds with 3 or more benzyl rings (I1), and a boolean attribute identifying compounds which are acanthryles (Ia). Ia, I1, logP and LUMO are relevant properties in determining mutagenicity.

3.4 Bongard Problems

Bongard problems[1] were introduced as an artificial domain in the field of pattern recognition. A simplified form of Bongard problems has been used as an artificial domain in the field of ILP[13]. We use a similar form of Bongard problems for our artificial domain experiments. We use a Bongard problem generator to generate datasets as shown in Figure 1. Each dataset consists of a set of positive and negative examples. Each example consists of a number of simple geometrical objects placed inside one another. The task is to determine the particular set of objects, their shapes and their placement which can correctly distinguish the positive examples from the negative ones.

4. STRUCTURALLY LARGE CONCEPTS

In this section we present the experiments, results and the analysis on the comparison of the graph-based and logic-based approaches while learning structurally large concepts.

4.1 Experiments on the Mutagenesis Dataset

In order to compare the performance of the approaches while learning large structural concepts we ran Subdue and CProgol on the Mutagenesis dataset. Since we intended to compare the ability of the approaches to learn large structural concepts, both the relational learners were provided

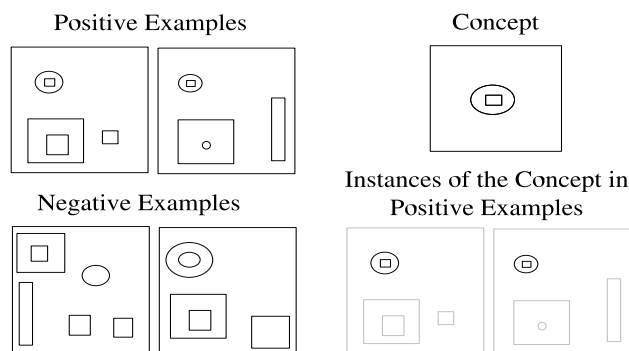


Figure 1: A Bongard Problem.

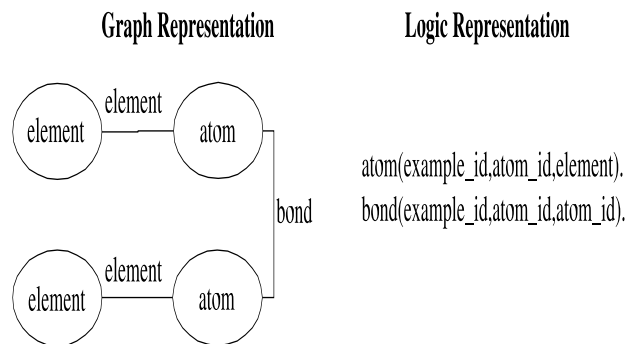


Figure 2: Representation of the Mutagenesis dataset while comparing the ability to learn structurally large concepts.

only with the basic information of the atoms, the elements and the bonds without any other information or background knowledge. This is shown in Figure 2. The relational learners are not provided with any additional information or any form of background knowledge, because we intended to compare the ability to learn large structural concepts. The introduction of any additional information or background knowledge would prevent this from happening. If systems were provided with the partial charge on the atoms and background knowledge to learn ranges, the systems would learn ranges on partial charges which would contribute to the accuracy. This would make it difficult to analyze how the approaches compare while learning structurally large concepts. Hence the partial charge information and the background knowledge to learn ranges was not given to either system. The atom type and bond type information was also not provided to either system. The reasoning behind doing so is that we view the atom type and bond type information as a propositional representation of relational data. Such information allows the relational learners to learn propositional representations of relational concepts instead of the true relational concept. Consider for example the rule found by CProgol on the Mutagenesis dataset[15], `atom(A,B,c,195,C)`. This rule denotes that compounds with a carbon atom of type 195 are mutagenic. The atom type 195 occurs as the atom shared by 3 fused rings 6 member rings. Therefore all

Table 1: Results on Mutagenesis Dataset while Comparing the Ability to learn Structurally Large Concepts

	CProgol	Subdue
Training Set Accuracy	60.00%	86.00%
Training Set Runtime	2010s	1876s
10-fold CV Accuracy	61.74%	81.58%
10-fold CV Runtime (average)	1940s	2100s
CProgol - Subdue, $\Delta\text{Error} \pm \sigma$	20.84% $\pm$ 12.78%	
CProgol - Subdue, Confidence	99.94%	

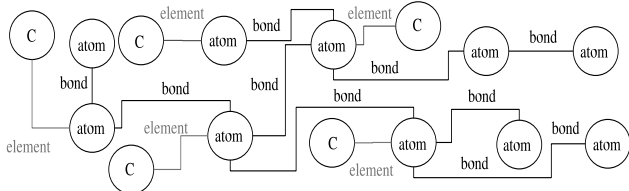


Figure 3: Rule Discovered by Subdue on the Mutagenesis Dataset while learning structurally large concepts.

compounds with 3 fused 6 member rings are labeled active. It is interesting to note that a rule involving 15 relations (3 fused 6 member rings) has been learned by learning a single relation. Learning such a rule has allowed CProgol to learn a propositional representation of a relational concept rather than the true relational concept. Providing atom type and bond type information would allow both systems to learn propositional representations of structurally large relational concepts rather than the true relational concepts. We do not consider the learning of such concepts equivalent to the learning of structurally large relational concepts. We therefore do not provide either system with the atom type and bond type information. The results of the experiment are shown in Table 1. For the training set, the accuracy for one run on the entire dataset and the learning time are shown. For 10-fold cross validation (CV), average learning time over 10 folds is shown.

The results show that Subdue performs significantly better than CProgol. Subdue learns 17 graphs representing 17 rules. One of the rules discovered by Subdue is shown in Figure 3. This rule has an accuracy of 76.72% and coverage of 81.15%. The hypotheses learned by CProgol mostly comprised of a single atom or bond predicate. The accuracy achieved by CProgol is comparable to that of random guessing. These results give a strong indication that a graph-based approach can perform better than a logic-based approach when learning structurally large concepts.

4.2 Artificial Domain Experiments

We performed additional experiments using artificially generated Bongard problems to reinforce the insights from the experiments on the Mutagenesis dataset. Figure 4 shows the representations used for Subdue and CProgol. We systematically analyzed the performance of Subdue and CProgol on artificially generated Bongard problems with increased number of objects in the concept and increased number of objects in the examples. In the first experiment, the number of objects in the Bongard concept was varied from

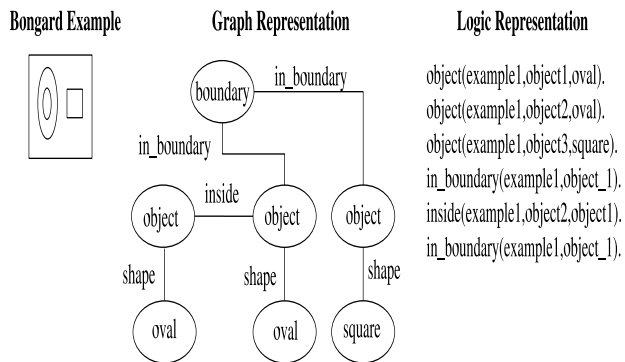


Figure 4: Representation for Bongard Problems.

5 to 35. The number of additional objects in each example (objects which are not a part of the concept) were kept constant at 5. For every concept size from 5 to 35, 10 different concepts were generated. For each of the 10 concepts a training set and a test set of 100 positive and 100 negative examples was generated. CProgol and Subdue were run on the training sets and were tested on the test sets. Figure 5 (a) shows the average accuracy achieved by CProgol and Subdue on 10 datasets for every concept size ranging from 5 to 35. It is observed that Subdue clearly outperforms CProgol. In order to further analyze the performance of the systems we reran the same experiment but in this case the systems were iteratively given increased resources (this was achieved by varying the 'nodes' parameter in CProgol and the 'limit' parameter in Subdue) so that we could determine the number of hypotheses each system explored before it learned the concept (a cutoff accuracy of 80% was decided). Figure 5 (b) shows the number of hypotheses explored by each system so as to achieve an accuracy of 80% (this experiment was only performed for concept size varying from 5 to 18 as a significantly large amount of time was required). A snapshot of the experiment (Accuracy vs. Number of Explored Hypotheses) for concept size 10 is shown in Figure 5 (c). The results show that CProgol explores a larger number of hypotheses than Subdue.

A similar experiment for increased example size was performed where the concept size was kept constant at 5 and the example size was varied from 10 to 35. Figure 5 (d) shows the average accuracy achieved by CProgol and Subdue on 10 datasets for every example size ranging from 10 to 35. Figure 5 (e) shows the hypotheses required to be explored to learn the concept (a cutoff accuracy of 80% was decided) determined by iteratively increasing the resources for each system. A snapshot of the experiment (Accuracy vs. Number of Explored Hypotheses) for example size 15 is shown in Figure 5 (f). Again, the results show that CProgol explores a larger number of hypotheses than Subdue.

4.3 Analysis

Here we attempt to explain the empirical results based on the algorithmic differences between the two approaches. An analysis of CProgol indicates that it first generates a most-specific clause from a randomly selected example using the mode definitions. Mode definitions together with the background knowledge form a user-defined model for gener-

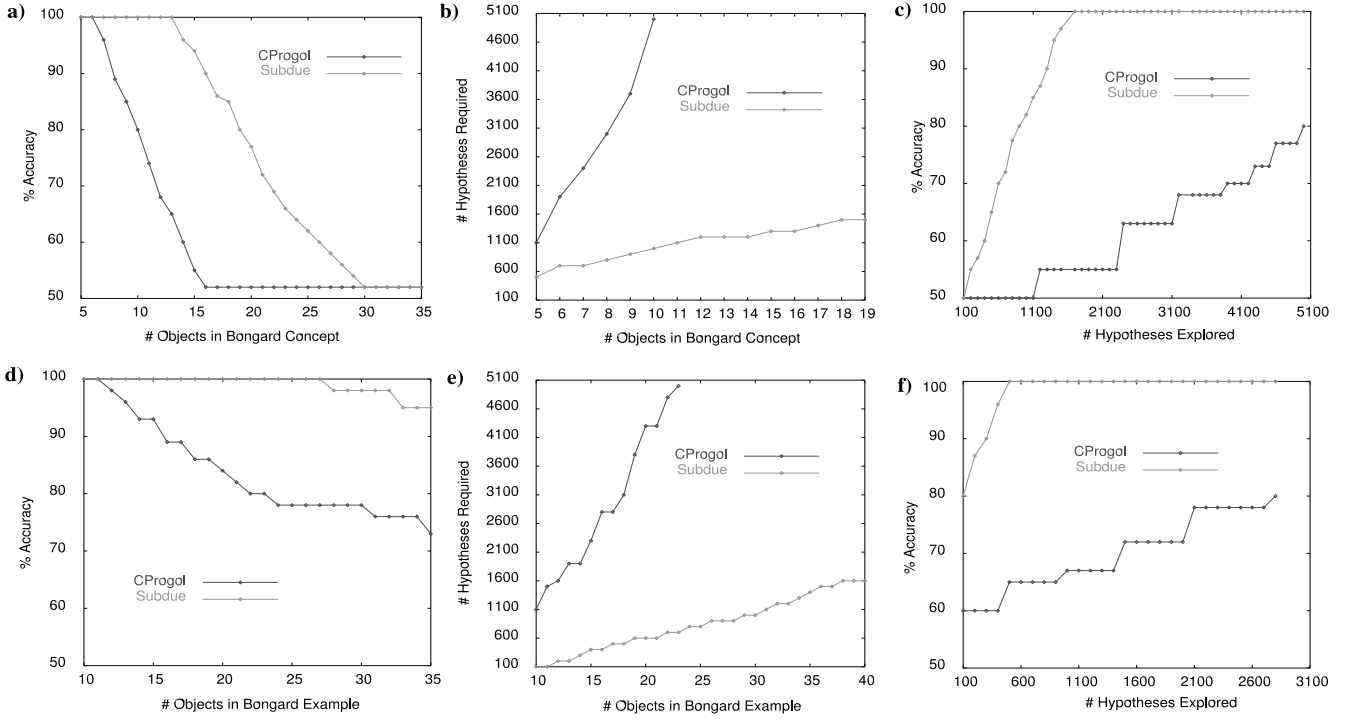


Figure 5: Results on artificial Bongard problems while comparing the ability to learn structurally large concepts

ation of candidate hypotheses. After generation of the most-specific clause, CProlog performs a general-to-specific search in the bounded θ -subsumption lattice guided by the mode definitions. The most general hypothesis is the empty clause and the most specific hypothesis is the clause generated in the previous step. The process of hypothesis generation is affected more by the mode definitions and the background knowledge than the examples, firstly because a single example is used to construct the most specific clause, and secondly because the mode definitions have a major effect on the process of hypothesis generation. Thus, CProlog makes more use of the mode definitions and background knowledge and less use of the examples. This observation about CProlog can be partially generalized to other logic-based approaches like top-down search of refinement graphs[12], inverse resolution[10] and relative least general generalization[11]. An analysis of Subdue indicates that hypotheses are generated only on the basis of the examples. The candidate hypotheses are generated by extending the sub-graph by an edge and a vertex or just an edge in all possible ways as in the examples. As Subdue generates the hypotheses only on the basis of the examples, it is more example driven. This observation about Subdue can be partially generalized to other graph-based systems like FSG[6], AGM[5] and gSpan[16], because there is more use of the examples and less use of the model. Subdue tends to explore the hypothesis space more efficiently because they use only the examples to generate candidate hypotheses, and thus can search a larger portion of the smaller hypothesis space with a given amount of resources, which is essential in learning structurally large relational concepts.

5. SEMANTICALLY COMPLICATED CONCEPTS

In this section we present the experiments, results and the analysis on the comparison of graph-based and logic-based approaches while learning semantically complicated concepts.

5.1 Experiments on the Mutagenesis Dataset

In order to compare the performance of the approaches while learning semantically complicated concepts, we ran Subdue and CProlog on the Mutagenesis dataset. Each system was provided with background knowledge so that numerical ranges could be learned. For CProlog this was achieved by introducing Prolog based background knowledge. For Subdue this was achieved by explicitly instantiating the background knowledge, i.e., additional structure was added to the training examples. This is shown in Figure 6. The results of this experiment are shown in Table 2. The results indicate that CProlog uses the background knowledge and shows an improved performance while Subdue has achieved a lower accuracy than what it achieved without the background knowledge. These results give a strong indication that a logic-based approach performs better than a graph-based approach when learning semantically complicated concepts.

Additional experiments (not reported in this paper) were performed with Subdue using various other forms of explicit instantiation to learn ranges. In all the experiments, Subdue did not learn ranges effectively, suggesting future investigation of a non-instantiation-based approach to introducing

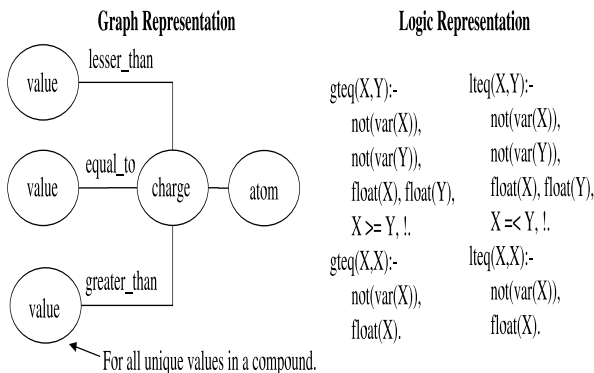


Figure 6: Representation of the Mutagenesis dataset while comparing the ability to learn semantically complicated concepts.

Table 2: Results on the Mutagenesis Dataset while Comparing the Ability to learn Semantically Complicated Concepts

	CProgol	Subdue
Training Set Accuracy	67.00%	64.00%
Training Set Runtime	1180s	2876s
10-fold CV Accuracy	66.53%	63.91%
10-fold CV Runtime (average)	1330s	2900s
CProgol - Subdue, $\Delta\text{Error} \pm \sigma$	2.16% $\pm$ 3.5%	
CProgol - Subdue, Confidence	95.77%	

background knowledge.

5.2 Analysis

Subdue explores only those hypotheses which are explicitly present in the examples. For hypotheses to be explicitly present in the examples, it is essential that the semantically complicated concepts that have to be learned be explicitly instantiated in the examples. An example of this is the explicit instantiation in the Mutagenesis experiment for Subdue to learn ranges. The drawbacks of the data-driven approach are that explicit instantiation is cumbersome in most cases and also that explicit instantiation is not a generalized methodology to learn complicated semantic concepts. For example, suppose a domain expert were to suggest that the ratio of the number of carbon atoms to the number of hydrogen atoms in a molecule has an effect on the mutagenicity. CProgol with some added background knowledge could use this information to classify the molecules. Subdue on the other hand would require making changes to the representation such that the pattern would be found in terms of a graph. CProgol allows the exploration of hypotheses through implicitly defined background knowledge rather than explicit instantiation in the examples. This is essential in learning semantically complicated multi-relational concepts.

6. BACKGROUND KNOWLEDGE

In this section we present the experiments, results and analysis on the comparison of graph-based and logic-based approaches while utilizing hypothesis space condensing background knowledge.

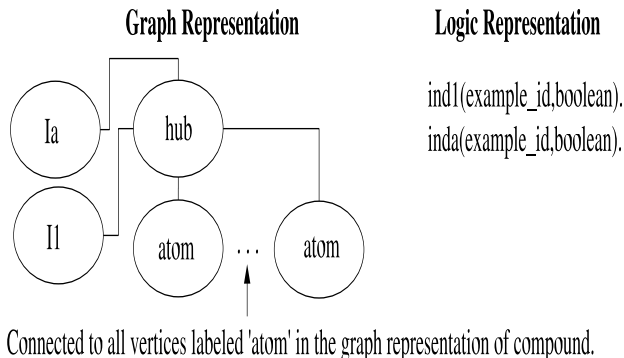


Figure 7: Representation of the Mutagenesis dataset while comparing the ability to utilize background knowledge in the form of indicator variables.

Table 3: Results on the Mutagenesis Dataset while Comparing the Ability to Utilize Background Knowledge in the Form of Indicator Variables

	CProgol	subdue
Training Set Accuracy	82.00%	80.00%
Training Set Runtime	960s	848s
10-fold CV Accuracy	78.91%	77.39%
10-fold CV Runtime (average)	810s	878s
CProgol - Subdue, $\Delta\text{Error} \pm \sigma$	1.52% $\pm$ 11.54%	
CProgol - Subdue, Confidence	31.38%	

6.1 Experiments on the Mutagenesis Dataset

In order to compare the ability of graph-based and logic-based approaches to utilize background knowledge, each system was provided with the background knowledge indicating the presence of benzyl rings (I1) and identifying compounds which are acenethryles (Ia). This is shown in Figure 7. Table 3 shows the results of this experiment. The results indicate that CProgol uses the background knowledge and show an improved performance while Subdue has achieved a lower accuracy than what it achieved without the background knowledge. While the results are not statistically significant, they give an indication that logic-based approaches tend to utilize background knowledge more effectively than graph-based approaches. The background knowledge provided to the systems was in the form of boolean indicator variables. In order to compare the ability of graph-based and logic-based approaches to utilize more complicated forms of background knowledge, each system was provided with the background knowledge indicating certain generic chemical concepts like benzene rings, nitro groups, etc. This is shown in Figure 8. Table 4 shows the results of this experiment. The results indicate that neither system uses the background knowledge or shows an improved performance.

6.2 Artificial Domain Experiments

We performed additional experiments using artificially generated Bongard problems to reinforce the insights from the experiments on the Mutagenesis dataset. Figure 9 shows the representations used for Subdue and CProgol. We systematically analyzed the performance of Subdue and CProgol on artificially generated Bongard problems with in-

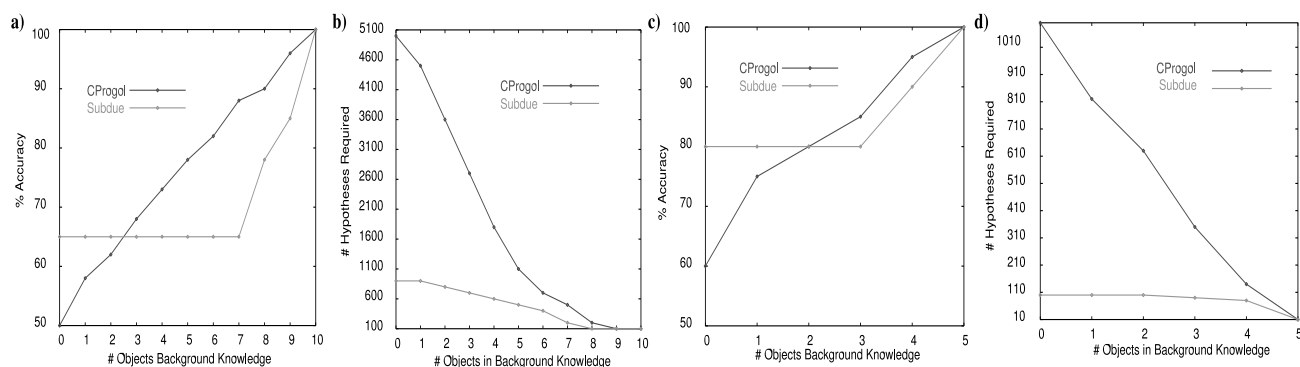


Figure 10: Results on artificial Bongard problems while comparing the ability to utilize background knowledge.

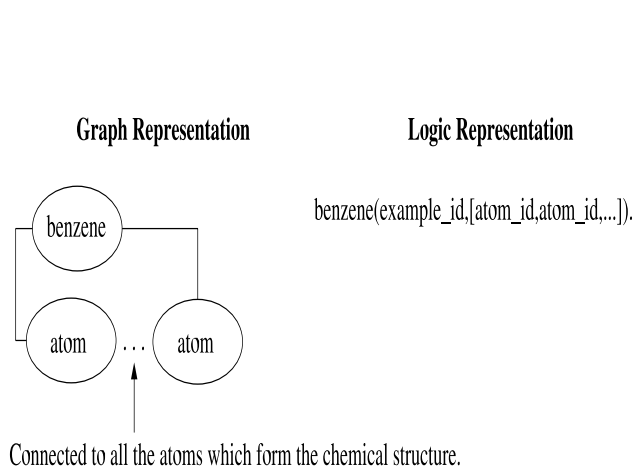


Figure 8: Representation of the Mutagenesis dataset while comparing the ability to utilize background knowledge indicating generic chemical concepts.

Table 4: Results on the Mutagenesis Dataset while Comparing the Ability to Utilize Background Knowledge Indicating Generic Chemical Concepts

	CProgol	Subdue
Training Set Accuracy	62.00%	64.00%
Training Set Runtime	2130s	1910s
10-fold CV Accuracy	61.74%	63.84%
10-fold CV Runtime (average)	2212s	2010s
CProgol - Subdue, $\Delta\text{Error} \pm \sigma$	1.74% $\pm$ 25.12%	
CProgol - Subdue, Confidence	16.86%	

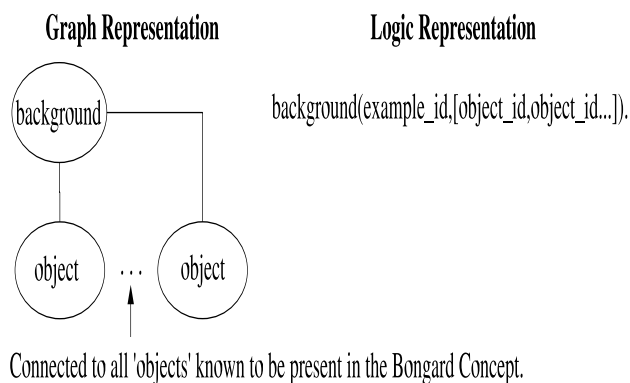


Figure 9: Representation for introducing background knowledge in Bongard Problems.

creased amount of background knowledge while learning a large concept (more objects in the concept) and with increased amounts of background knowledge while learning a concept from a large example (more objects in each example). In the first experiment, for a concept of size 10 and additional objects in each example equal to 5, 10 concepts were generated. For each of these concepts a training set and test set of 100 positive and 100 negative examples were generated. Figure 10 (a) shows the accuracies achieved by Subdue and CProgol (Note that both the systems were given less resources than the experiments in Section 4.3 so that the effect of background knowledge could be analyzed). Figure 10 (b) shows the hypotheses required by each system to learn the concept (a cutoff accuracy of 80% was decided). CProgol shows a larger improvement in performance than Subdue after the introduction of background knowledge. A similar experiment for a concept of size 5 and example size of 15 was performed. Figure 10 (c) shows the accuracies achieved by Subdue and CProgol (Note that both the systems were given less resources than the experiments in Section 4.3 so that the effect of background knowledge could be analyzed). Figure 10 (d) shows the hypotheses required by each system to learn the concept (a cutoff accuracy of 80% was decided). Again, CProgol shows a larger improvement in performance than Subdue after the introduction of background knowledge.

6.3 Analysis

As mentioned previously, Subdue explores only those hypotheses which are explicitly present in the examples. Thus, in order to use background knowledge, it is essential for the background knowledge to be explicitly present in the examples. This is the reason why background knowledge is introduced in the form of a vertex which is connected to all the entities which comprise the background knowledge, as in the Bongard problems. In the case of CProgol, similar background knowledge is introduced in the form of the predicate, `background(example_id,[object_id, object_id..])`. Subdue generates candidate hypotheses by extending the sub-graph by an edge and a vertex or just an edge in all possible ways as in the examples. The resulting increase in hypothesis space to be explored by Subdue after the introduction of background knowledge is larger than that of CProgol which can generate candidate hypotheses by adding a background knowledge predicate in a single refinement. Subdue does provide an alternate way of introducing background knowledge by preprocessing the examples and compressing each of the user defined substructures which form the background knowledge into a single vertex. This technique is more efficient, but leads to information loss and Subdue will not be able to learn a concept that contains only a partial portion of the background knowledge substructure. Thus, CProgol utilizes background knowledge more effectively than Subdue.

7. CONCLUSIONS AND FUTURE WORK

We performed an experimental comparison of the graph-based multi-relational data mining system, Subdue, and the inductive logic programming system, CProgol. From this comparison we conclude that Subdue tends to explore the hypothesis space more efficiently which is essential in learning structurally large relational concepts. CProgol makes efficient use of background knowledge and is better at learning semantically complicated concepts. Subdue requires instantiated background knowledge and can only learn concepts which are explicitly instantiated into the examples. Subdue needs to achieve the ability to use background knowledge and learn semantically complicated concepts. CProgol needs mechanisms which explore the search space more efficiently. Developing methodologies for using existing graph-based and logic-based systems in combination is an immediate, but less efficient, way to learn structurally large and semantically complicated concepts. For example, the structurally complicated hypotheses learned by a graph-based system could then be used as background knowledge by a logic-based system.

The conclusions drawn for this case study provide initial insights about the differences between graph-based and logic-based MRDM. As the study was limited to one representative system of each approach, and a single real world domain, these conclusions cannot be generalised to graph-based and logic-based approaches to MRDM. A similar study involving more systems and more domains would be required to generate insights about the fundamental differences between graph-based and logic-based MRDM. We plan to pursue this in future.

8. ACKNOWLEDGEMENTS

This research is sponsored by the Air Force Research

Laboratory (AFRL) under contract F30602-01-2-0570. 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 AFRL or the United States Government.

9. REFERENCES

- [1] M. Bongard. *Pattern Recognition*. Spartan Books, 1970.
- [2] D. J. Cook and L. B. Holder. Substructure discovery using minimum description length and background knowledge. *J. Artif. Intell. Res. (JAIR)*, 1:231–255, 1994.
- [3] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Min. Knowl. Discov.*, 3(1):7–36, 1999.
- [4] S. Dzeroski. Multi-relational data mining: an introduction. *SIGKDD Explorations*, 5(1):1–16, 2003.
- [5] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD*, pages 13–23, 2000.
- [6] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1038–1051, 2004.
- [7] T. Matsuda, T. Horiuchi, H. Motoda, and T. Washio. Extension of graph-based induction for general graph structured data. In *PAKDD*, pages 420–431, 2000.
- [8] S. Muggleton. Inductive logic programming. *New Generation Comput.*, 8(4):295–, 1991.
- [9] S. Muggleton. Inverse entailment and progol. *New Generation Comput.*, 13(3&4):245–286, 1995.
- [10] S. Muggleton and W. L. Buntine. Machine invention of first order predicates by inverting resolution. In *ML*, pages 339–352, 1988.
- [11] S. Muggleton and C. Feng. Efficient induction of logic programs. In *ALT*, pages 368–381, 1990.
- [12] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
- [13] L. D. Raedt and W. V. Laer. Inductive constraint logic. In *ALT*, pages 80–94, 1995.
- [14] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific Publishing, 1989.
- [15] A. Srinivasan, S. Muggleton, M. J. E. Sternberg, and R. D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artif. Intell.*, 85(1-2):277–299, 1996.
- [16] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724, 2002.