Journal of Naval Science and Engineering
2015, Vol. 11, No.1, pp.19-33

USING GRAPHS IN MULTI RELATIONAL DATA MINING

Mahmut IGDE [1]
Yusuf KAVURUCU [2]
Alev MUTLU [3]

[1,2]Computer Engineering Department,
Turkish Naval Academy, Naval Sciences and Engineering Institute, Tuzla, Istanbul
[1] migde@dho.edu.tr, [2]ykavurucu@dho.edu.tr

[3] Computer Engineering Department,
Kocaeli University, Kocaeli
[3]alev.mutlu@kocaeli.edu.tr

Abstract

Multi-relational concept discovery aims to find the relational rules that best describe the target concept. In this paper, we present a graph-based concept discovery method in Multi-Relational Data Mining. Concept rule discovery aims at finding the definition of a specific concept in terms of relations involving background knowledge. The proposed method is an improvement over a state-of-the-art concept discovery system that uses both ILP and conventional association rule mining techniques during concept discovery process. The proposed method generates graph structures with respect to data that is initially stored in a relational database and utilizes them to guide the concept induction process. A set of experiments is conducted on data sets that belong to different learning problems. The results show that the proposed method has promising results in comparison to state of the art methods.

ÇOKLU İLİŞKİSEL VERİ MADENCİLİĞİNDE GRAFİKLERİN KULLANIMI

Özetçe

Çok ilişkili konsept keşfinin amacı hedef konsepti en iyi şekilde anlatabilen ilşkisel kuralları bulmaktır. Bu çalışma ile çok ilişkili veri madenciliğinde diyagram tabanlı konsept keşif metodundan bahsediyoruz. Konsept kural keşfi, arkaplan bilgilerini içeren ilişkileri gözönünde bulundurarak özel bir konsetin tanımını bulmayı hedefler. Anlatılan metot C^2D

konsept keşif sisteminin geliştirlmesi ile elde edilmiştir. C^2D konsept keşfi esnasında ILP ve geleneksel ortaklık kural madenciliği (APRIORI gibi) tekniklerini birlikte kullanır. Anlatılan sistem, isim olarak D-KKS(Diyagram tabanlı Konsept Keşif Sistemi), başlangıçta ilişkisel veritabanında tutulan verilere bağlı kalmak kaydı ile diyagram yapılarını oluşturur ve bu verileri kullanarak konsept çıkarsama sürecini yönlendirir.Farklı öğrenme problemleri ile alakalı veri setleri üzerinde testler yapılmıştır. Test sonuçları D-KKS'nin, bu alandaki diğer sistemler ve C^2D'ye nispeten umut verici olduğunu göstermektedir.

Keywords: *Concept Discovery, Graph, Path, MRDM, ILP*
Anahtar Kelimeler: *Konsept Keşfi, Grafik, , Yol, Çok İlişili Veri Madenciliği, Tümevaran Mantıksal Programlama*

1. INTRODUCTION

Due to increase of complex data usage in information systems, the amount of data collected in relational databases is also increasing. This increase forced the development of multi-relational learning algorithms that can be applied to directly multi-relational data stored in databases (Dzeroski-2003). Generally, first-order predicate logic is employed as the representation language for such learning systems. The learning systems, which find logical patterns valid for given background knowledge, have been investigated under a research area which is called Inductive Logic Programming (ILP) (ILP-Muggleton-1999).

Concept is defined as a set of frequent patterns that are embedded in the features of the concept instances in the form of relations among objects (Huchard, H.,R.,V.,2007). Concept discovery is the problem of learning definitions of a specific relation, called **target relation**, in terms of other relations provided as **background knowledge** (Learn-Logic-FOIL-Quinlan-1990). Concept discovery in relational databases is a predictive learning task. There is a specific target concept to be learned in the light of the past experiences. In ILP-based concept learning methods, logical patterns for the target concept are induced that are validated against the background facts. Association rule mining is a technique that is employed in the proposed algorithms for relational concept discovery. Association rule mining is finding frequent patterns, associations or correlations among sets of items or objects in

databases. Relational association rules are expressed as query extensions in first-order logic (Dehaspe, Toivonen-2001).

The concept discovery problem has extensively been studied by the ILP community with successful applications in several domains such as bioinformatics, engineering, and environmental sciences. Among several problems in concept discovery, a common problem faced by ILP-based concept discovery systems is the so called **local plateau problem** (Alphonse,Osmani,2008). In such cases classical operators of ILP that refine concept descriptors by one literal at a time are insufficient to improve the quality of the concept descriptors and the systems perform a blind search. To the best of our knowledge, graph-based approaches were first introduced to the concept discovery problem to solve this issue by Richards and Mooney (Richards,Mooney,1992). In their approach the refinement operators of ILP are upgraded to refine the concept descriptors by a set of literals such that arguments of the literals form a path.

Graph-based approach is another concept discovery method which is based on graph structure. Graph-based concept discovery methods can be classified into two main categories: Substructure-based approaches and path finding-based approaches (DAWAK-13-GRAPH). In a graph if a substructure is seen frequently then there should be a concept which constructs that substructure. This is the idea behind substructure based approach. On the other hand, path finding-based approaches (Gao,Z.,H.,2009) assume that a concept should appear as frequent and finite length paths that connect some arguments of positive target instances. Such approaches need to employ advanced indexing mechanisms to keep track of the paths.

In this work we propose a hybrid framework, namely G-CDS (Graph-based Concept Discovery System), for graph-based concept discovery. We employ directed, labeled graph where nodes represent target and background relations, and edges connect those nodes that have at least one common argument. The proposed approach inputs the data in relational format, generates a graph for each target instance, generates the summary graph, extracts concept definitions, and outputs concept descriptors in the form of

relations. Similar to substructure-based approaches, it groups similar relations and represents them as a single node. Similar to path finding-based approaches, it infers the concept descriptors by finding paths that connect relations. Different than substructure-based approaches, the proposed approach does not employ graph isomorphism algorithms, which are known to be NP, to group similar nodes but rather constructs the graph in a compressed form by executing SQL queries on the input data. Different than path finding-based approaches, the proposed method does not search for paths within the graph, but infers such paths while constructing the graph. The proposed method does not need to employ advanced indexing mechanism to store paths either, but keeps such information within the nodes.

A challenging problem of relational concept discovery is dealing with intractably large search space. Several systems have been developed employing various search strategies, language pattern limitations and hypothesis evaluation criteria, in order to prune the search space. However, there is a trade-off between pruning the search space and generating high-quality patterns. Major features of G-CDS are as follows:

1. Instead of strong declarative biases such as input-output modes, the information inside the relational database schema such as argument types and primary-foreign key relationships, a confidence-based pruning mechanism and APRIORI method (Agrawal,1996) are used to prune the search space similar to C^2D.

2. G-CDS directly works on relational databases without any requirement of negative instances.

3. Aggregate predicates are defined and incorporated into concept discovery process. In addition, a simple method was developed to handle comparison operators on numeric attributes, which generally accompany aggregate predicates.

The experimental results of G-CDS revealed promising performance on the quality of concept discovery in comparison with similar works

(Kavurucu, Expert Syst. Appl.,2009 - Kavurucu ,Knowledge-Based Systems, 2010).

This paper is organized as follows:
Section 2 presents the related work. Section 3 gives the preliminary information for graph-based concept discovery. Section 4 gives an overview of G-CDS. Section 5 presents the experiments to discuss the performance of G-CDS. Finally, Section 6 includes concluding remarks.

2. RELATED WORK

ILP-based concept discovery systems distinguish from each other in terms of the hypothesis formation technique, search direction, the need of mode declarations, allowing recursion and negated predicates in the body part. FOIL, PROGOL, ALEPH, WARMR, C^2D and CRIS are some of the well-known ILP-based systems in the literature. PROGOL (Inv-Ent-Progol-Muggleton-1995) is a top-down relational ILP system, which is based on inverse entailment. A bottom clause is a maximally specific clause, which covers a positive example and is derived using inverse entailment. PROGOL extends clauses by traversing the refinement lattice. ALEPH (ALEPH manual,1999) is similar to PROGOL, whereas it is possible to apply different search strategies and evaluation functions.

Design of algorithms for frequent pattern discovery has become a popular topic in data mining. Almost all algorithms have the same level-wise search technique known as APRIORI algorithm. WARMR (Mine-AR-Dehaspe-Raedt-1997) is a descriptive ILP system that employs Apriori rule to find frequent queries having the target relation. C^2D (Kavurucu, Expert Syst. Appl.,2009) and CRIS (Kavurucu ,Knowledge-Based Systems, 2010) are two ILP-based concept discovery systems behind which the basic motivation is to develop a system that facilitates concept discovery by non-expert users for the data stored in relational databases. They are similar to ALEPH as both systems produce concept definition from given target. WARMR is another similar work in a sense that, both systems employ Apriori-based searching methods. However, unlike ALEPH and WARMR, they do not need input/output mode

declarations. They only require type specifications of the arguments, which already exist together with relational tables corresponding to predicates. ALEPH and WARMR can use indirectly related relations and generate transitive rules only with using strict mode declarations. However, in C^2D and CRIS, transitive rules are generated by using indirectly related relations without the guidance of mode declarations. Most of the ILP-based systems require negative information, whereas C^2D and CRIS directly works on databases which have only positive data. Finally, they use a novel confidence-based hypothesis evaluation criterion and search space pruning method. Graph-based concept discovery systems can be classified as substructure-based approaches and path finding-based approaches. SubdueCL (Gonzalez, Holder, 2001) represent data as a directed, labeled graph. In that graph, nodes store arguments of the facts, and labeled edges are the relation names connecting the arguments of the facts. In SubdueCL, substructures are evaluated according to the number of positive and negative target instances they explain. Another concept learning system based on substructure discovery is Graph Based Induction (GBI) (Yoshida,Motoda, 1995). It employs colored digraph as the representation framework where colors attached to the nodes represent the attributes of the facts. GBI examines each connected pair of nodes, and merges the frequent typical ones. The final merged substructures are labeled as concepts.

Relational Pathfinding (Richards,Mooney,1992) is one of the earliest path finding-based approaches which aims to overcome the local plateau problem of ILP-based concept discovery systems. In Relational Pathfinding, similar to SubdueCL, nodes represent fact arguments. Edges are labeled after the relation names and connect such pairs of nodes that they form a fact. It employs bidirectional breadth first search to discover the concept descriptors. Relational Paths Based Learning (RPBL) (Gao,Z.,H.,2009) is yet an other concept discovery system based on path finding. In RPBL, nodes represent binary facts, and edges connect nodes that share some arguments in common. To learn recursive concept descriptors, extended version of RPBL treats the target instances also as background knowledge. To apply domain theories into the learning process, they extend the graph in accordance with domain theories, i.e. by connecting nodes that hold with the domain theories. The proposed

approach is similar to substructure-based approaches as it works on a compressed graph. Different than such studies, the graph is not compressed to find concept descriptors but to provide a compact representation of the data. Similar to path finding-based approaches it represents the concept descriptors as a path that connects arguments of some target instances. Different than such studies it does not look for paths on a already built graph, but discovers such paths while constructing the graph.

3. G-CDS: GRAPH-BASED CONCEPT DISCOVERY SYSTEM

In this section, we present the hybrid graph-based concept discovery process, and list the distinguishing properties of the proposed method from state of the art methods. We employ the **elti** data set given in Table 1 as a running example throughout this section. In the data set, predicate **e** stands for the **elti** relation, **h** stands for the **husband** relation, **w** stands for the **wife** relation, and **b** stands for the **brother** relation. All arguments are of type **person**. The **elti** relation is the concept to be learned, **husband**, **brother**, and **wife** are the background relations. **elti** is a kinship relation in Turkish and represents two people if they are wives of two brothers.

Table 1. The elti data set

Target Instance		
t1:e(cemile,ayse)	**t2:e(cemile,ayten)**	**t3:e(nalan,bedriye)**
Background Data		
b1: b(mehmet,ismail)	**b2: b(mehmet,ali)**	**b3: b(sadullah,yildirim)**
b4: h(sadullah,nalan)	**b5: h(ali,ayse)**	**b6: h(yildirim,bedriye)**
b7: h(mehmet,cemile)	**b8: h(ismail,ayten)**	**b9 : w(bedriye,yildirim)**
b10: w(ayten,ismail)	**b11: w(ayse,ali)**	**b12: w(nalan,sadullah)**
b13: w(cemile,mehmet)		

3.1 The Algorithm

The proposed method, namely G-CDS, takes a set of target instances, a set of background data, minimum support, minimum confidence, and maximum rule length as input data and discovers concept rules that describe

the target relation. The target instances and the background data are initially stored in a relational database.

The proposed method is composed of following components for each target instance (totally, there will be n graphs, where n is the number target instances):

Initialization: In this step, a graph is created with a single root node. The root node is labeled with target instance value (e.g. e(cemile, ayse)).

Expansion: The background facts which are related to selected target instance are selected from the database and the initial graph is expanded with adding nodes to the root node for each related background instance. Figure 1 represents the graph at the end of this step.

Check for Indirectly Related Instances: In this step, the indirectly related facts that are chosen among the unrelated facts in database are added to the graph. A background fact is indirectly related to target instance if it has a common argument not with the root node, but with the related facts. This process is repeated according to the maximum length parameter defined at the beginning of the overall process. Figure 2 represents the final graph with respect to selected target instance.

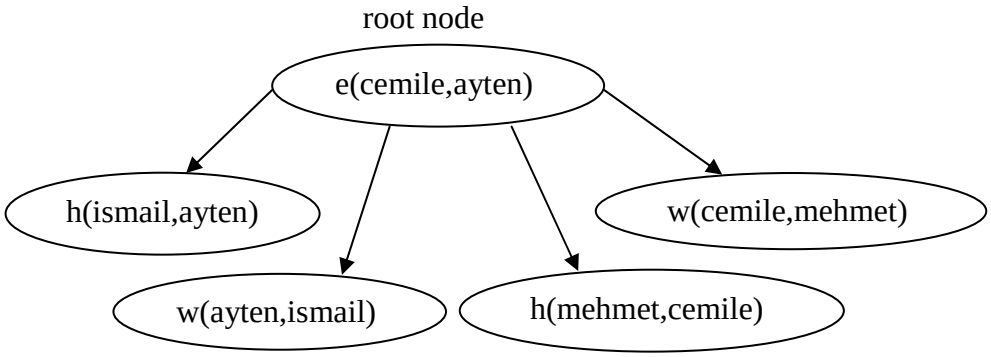


Fig. 1. The selected target instance and related background facts

Table 2. Example Support and Confidence Queries

Supp. = C1/C2	C1:SELECT COUNT DISTINCT (e.arg1, e.arg2) FROM elti e, husband h WHERE e.arg1 = h.arg2 AND e.arg2 = h.arg1
	C2:SELECT COUNT DISTINCT(e.arg1, e.arg2) FROM elti e;
Conf. = C3/C4	C3:SELECT COUNT DISTINCT(h.arg1, h.arg2) FROM elti e, husband h WHERE e.arg1 = h.arg2 AND e.arg2 = h.arg1
	C4:SELECT COUNT DISTINCT(h.arg1, h.arg2) FROM husband h

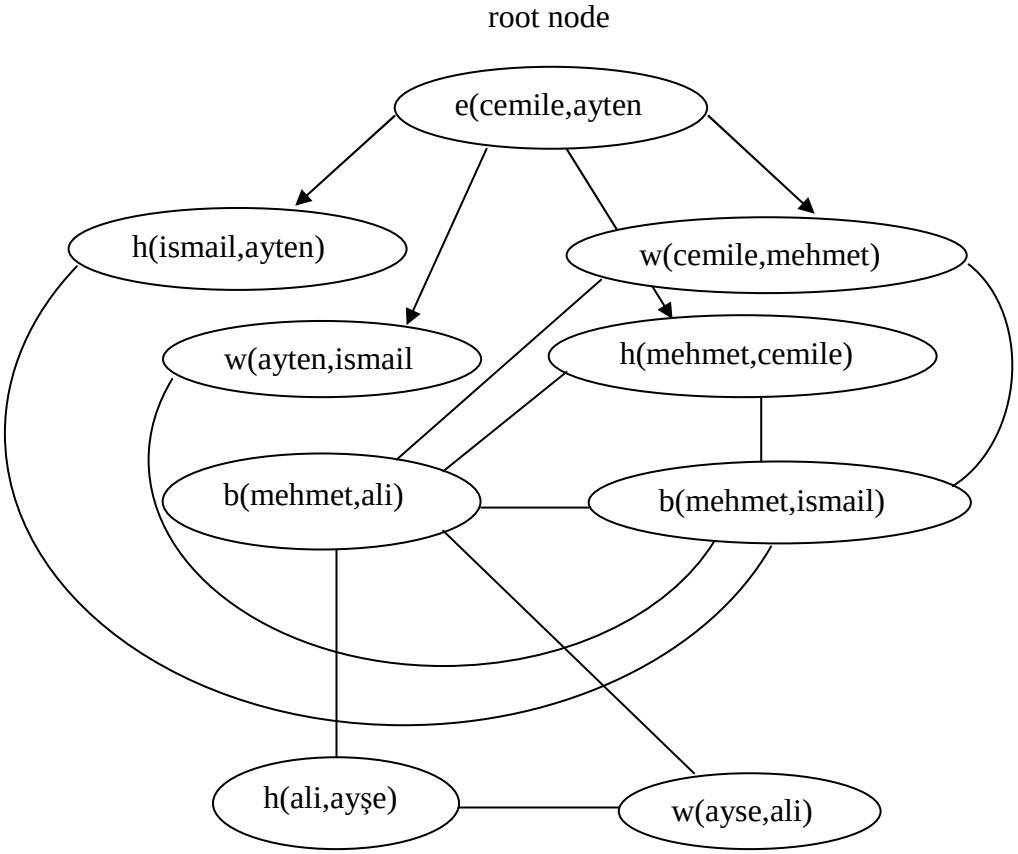
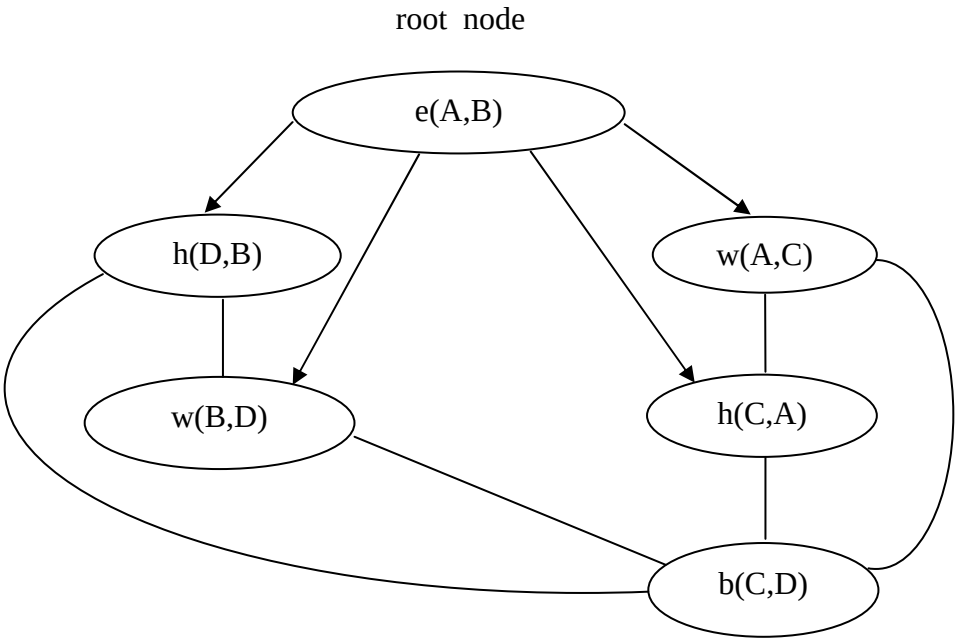


Fig. 2. The graph with depth=3 for the selected target instance

Graphs comparison for template construction: In this step, the graphs we constructed for all the target instances one by one are compared to each other and a template graph which is acceptable for all of them is declared. The paths which have the same relations from the beginning to the end in all graphs are accepted as a final path and added to our solution set. In Figure 3, our solution graph is shown.

p1: e(A,B), w(A,C), b(C,D), b(C,D)
p2: e(A,B), w(A,C), b(C,D), w(E,D)
p3: e(A,B), w(A,C), b(C,D), h(D,E)
p4: e(A,B), h(C,A), b(C,D), w(E,D)
p5: e(A,B), h(C,A), b(C,D), h(D,E)
p6: e(A,B), w(B,C), b(D,C), b(D,E)
p7: e(A,B), w(B,C), b(D,C), h(D,A)
p8: e(A,B), w(B,C), b(D,C), w(A,D)
p9: e(A,B), w(B,C), b(D,C), h(C,B)
p10:e(A,B), h(C,B), b(D,C), w(B,C)
p11:e(A,B), h(C,B), b(D,C), w(A,D)
p12:e(A,B), h(C,B), b(D,C), h(D,A)
p13:e(A,B), h(C,B), b(D,C), b(D,C)

Fig. 3. The Template Graph having frequent edges



Update Variables : When we constructed our solution graph, variables are to be updated because of the differences, resulted of they are coming from different graphs. Finally, we can extract the rules as a candidate solution set:
c1: e(A,B):- w(A,C), b(C,D), w(B,D)
c2: e(A,B):- w(A,C), b(C,D), h(D,B)
c3: e(A,B):- h(C,A), b(C,D), w(B,D)
c4: e(A,B):- h(C,A), b(C,D), h(D,B)
c5: e(A,B):- w(B,D), b(C,D), h(D,B)
c6: e(A,B):- w(B,D), b(C,D), w(A,C)
c7: e(A,B):- w(B,D), b(C,D), h(C,A)
c8: e(A,B):- w(B,D), h(D,B), e(A,B)
c9: e(A,B):- h(D,B), b(C,D), w(A,C)
c10:e(A,B):- h(D,B), b(C,D), h(C,A)
c11:e(A,B):- h(D,B), b(C,D), w(B,D)

Evaluation and pruning: In this step support and con dence values of the current concept descriptors are calculated. To calculate these values, current concept descriptors are translated into SQL queries and these queries are run against the database. Please note that these concept descriptors are indeed the paths that connect the tail nodes to the source node, and this information is stored within each tail node. In Table 2, we provide support and confidence queries for elti(A, B):- husband(B, A).

Covering: In this step target instances explained by the solution clauses are marked as covered. If the number of the remaining uncovered target instances is below minimum support ┐ #target instances the concept induction process terminates, else restarts with the initialization step.

4.EXPERIMENTAL RESULTS

4.1 Learning Recursive Rules

One of the interesting test cases that we have used is a complex family relation, ``same-generation'' learning problem. In the data set, 344 pairs of

actual family members are given as positive examples of **same-generation (sg)** relation. Additionally, 64 background facts are provided to describe the **parental (p)** relationships in the family. We set the support threshold as 0.3, confidence threshold as 0.6 and maximum depth as 3.

G-CDS finds the following clauses (similar to C^2D/CRIS) for this data set:

sg(X, Y) :- p(Z, X), p(U, Y), sg(Z, U).

sg(X, Y) :- p(Z, Y), p(U, X), sg(Z, U).

sg(X, Y) :- p(Z, X), p(Z, Y).

For this data set, ALEPH and PROGOL cannot find a solution under default settings. Under strong mode declarations and constraints, ALEPH finds the following hypothesis:

sg(X, Y) :- p(Z, X), p(Z, Y).

sg(X, Y) :- sg(X, Z), sg(Z, Y).

sg(X, Y) :- p(Z, X), sg(Z, U), p(U, Y).

However, PROGOL can only find ``sg(X, Y) :- sg(Y, Z), sg(Z ,X)." as a solution. The experiment shows that, G-CDS can find the correct hypothesis set for the same generation problem whereas ALEPH and PROGOL cannot.

4.2 Constructing Transitive Rules Under the Existence of Indirectly Related Facts

Michalski's trains problem (Michalski-1997) is a typical case in which the most background facts are indirectly related to target instances. In this data set, the target relation **eastbound(train)** is only related with **has_car(train,**

car) relation. The other background relations have an argument of type **car** and are only related with **has_car** relation.

The **eastbound** relation has 5 records which are **east1, east2, east3, east4, east5**. The target relation has one parameter and its type is train. One of the background relations (has_car) has only related column type and facts. The other background relations are not related. By adding indirectly related facts in the discovery process, C^2D finds the following rule for this data set:

eastbound(A) :- has_car(A, B), closed(B). (s=5/5, c=5/7).

The best rule for this data set is actually different than what C^2D found. Because of the pruning mechanisms of C^2D, it is unable to find the best rule. However, G-CDS generates five graphs and the summary graph and by defining maximum rule length as three, it finds the following rule:

eastbound(A) :- has_car(A, B), closed(B), short(B).(s=5/5,c=5/5).

5. CONCLUSION

In this work we present a hybrid graph-based concept discovery for data stored in a multi-relational database. The method is hybrid, as it is similar to substructure-based approaches it is looks for similar nodes in graphs, and is similar to path-finding methods as it extracts concept descriptors by traversing a summary graph. The experimental results show that the proposed method is capable of inducing correct concept descriptors for datasets that belong to different learning problems. As a future work, we plan to further investigate the performance of G-CDS on datasets such as Mesh (Dolsak,Muggleton,1992) and Mutagenesis (Srinivasan,1994). The proposed approach is well suited for parallel execution, hence another future direction is parallelizing the proposed method.

REFERENCES

[1] Dzeroski, S.: Multi-relational data mining: an introduction. SIGKDD Explorations 5(1) (2003) 1-16

[2] Muggleton, S.: Inductive Logic Programming. In: The MIT Encyclopedia of the Cognitive Sciences (MITECS). MIT Press (1999)

[3] Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Ann. Math. Artif. Intell. 49(1-4) (2007) 39-76

[4] Quinlan, J.R.:Learning logical definitions from relations. Mach. Learn. 5(3) (1990) 239-266

[5] Dehaspe, L., Toivonen, H.: Discovery of relational association rules. In Dzeroski S., Lavra¶ c, N., eds.: Relational Data Mining. Springer-Verlag (September 2001) 189-212

[6] Alphonse, E, Osmani, A.: On the connection between the phase transition of the covering test and the learning success rate in ilp. Machine Learning 70(2-3) (2008) 135-150

[7] Mutlu, A., Karagoz, P.: A hybrid graph-based method for concept rule discovery. In: Data Warehousing and Knowledge Discovery - 15th International Conference, DaWaK 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings. (2013) 327-338

[8] Gao, Z., Zhang, Z., Huang, Z.: Learning relations by path finding and simultaneous covering. In: CSIE (5). (2009) 539-543

[9] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996) 307-328

[10] Kavurucu, Y., Senkul, P., Toroslu, I.H.: Ilp-based concept discovery in multirelational data mining. Expert Syst. Appl. 36(9) (November 2009) 11418-11428

[11] Kavurucu, Y., Senkul, P., Toroslu, I.H.: Concept discovery on relational databases: New techniques for search space pruning and rule quality improvement. Knowledge-Based Systems 23(8) (December 2010) 743-756

[12] Muggleton, S.: Inverse entailment and Progol. New Generation Computing, Special issue on Inductive Logic Programming 13(3-4) (1995) 245-286

[13] Srinivasan, A.: The Aleph Manual (1999)

[14] Dehaspe, L., Raedt, L.D.: Mining association rules in multiple relations. In: ILP'97:Proceedings of the 7th International Workshop on Inductive Logic Programming, London,UK, Springer-Verlag(1997) 125-132

[15] Gonzalez, J.A., Holder, L.B., Cook, D.J.: Graph-based concept learning. In:FLAIRS Conference. (2001) 377-381

[16] Yoshida, K., Motoda, H.: Clip: Concept learning from inference patterns. Artif.Intell. 75(1) (1995) 63-92

[17] Richards, B.L., Mooney, R.J.: Learning relations by path finding. In: AAAI. (1992) 50-55

[18] Michalski, R., Larson, J.: Inductive inference of vl decision rules. In: Workshop on Pattern-Directed Inference Systems. Volume 63., Hawaii, SIGART Newsletter, ACM (1997) 33-44

[19] Dolsak, B., Muggleton, S.: The application of inductive logic programming to finite-element mesh design. In Muggleton, S., ed.: Inductive Logic Programming. Academic Press (1992) 453-472

[20] Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.: Mutagenesis: Ilp experiments in a non-determinate biological domain. In: Proceedings of the 4th international workshop on inductive logic programming. Volume 237., Citeseer (1994) 217-232