

Guiding Search in Relational Pathfinding-based Concept Discovery via Bivariate Statistical Methods

Alev Mutlu*¹, Furkan Goz², Hatice Zeren³

Submitted: 04/11/2018 Accepted : 16/07/2019

Abstract: Relational pathfinding-based systems learn concept descriptors by extending candidate concept descriptors by one literal at a time. As such learning systems usually deal with large search spaces, choosing literals to extend candidate concept descriptors becomes an essential issue. In this study we empirically analyze applicability of three bivariate statistical methods namely, frequency ratio, hazard index, and weight of evidence, as heuristics to choose literals to extend candidate concept descriptors. 10-fold experiments conducted on three benchmark datasets showed that frequency ratio, hazard index, and weight of evidence were able to reduce the space and hence provided speedups when compared to extending candidate concept descriptors by a randomly chosen literal. Moreover, the heuristic-based settings provided improved predictive accuracy.

Keywords concept discovery, frequency ratio, hazard index, heuristic, relational-path, weight of evidence

1. Introduction

Concept discovery is a multi-relational data mining problem concerned with inducing definitions of a relation, called *target relation*, in terms of related knowledge, called *background knowledge* [1]. Target relation may contain instances that truly belong to the target relation, called positive examples, and instances that do not belong to the target relation, called negative examples. Background knowledge may be directly or indirectly related to the target instances and can either be expressed extensionally or intentionally. Goal in concept discovery is to find complete (explaining all of the positive target instances) and consistent (explaining none of the negative target instances) concept descriptors that satisfy some user-defined metrics such as minimum support and confidence, maximum rule length and obey certain mode declarations.

The problem has primarily been studied by Inductive Logic Programming (ILP) community. In ILP-based concept discovery systems relational data is represented within first order logic framework and logical operators are utilized for inductive inference [2]. Several ILP-based systems such as FOIL [3], ALEPH [4], and PROGOL [5] have been proposed with successful application in domains such as chemoinformatics [6], environmental sciences [7], and engineering [8].

More recently, the problem has been investigated from graph mining perspective and two distinct research directions have been established. The first direction of research focuses on substructure discovery and assumes substructures that involve positive target instances and satisfy certain user defined metrics are concept descriptors [9, 10]. The second direction of research focuses on paths and assumes that frequently appearing paths of finite length

that originate from positive target instances are concept descriptors [11,12].

In their basic settings, both substructure- and pathfinding-based systems rely on hill climbing strategy to build concept descriptors. More specifically, substructure-based systems extend candidate concept descriptors by one edge and vertex at a time, and pathfinding-based concept discovery systems extend candidate concept descriptors by one edge at a time. In case of very large graphs, this basic setting becomes inefficient and mechanisms to speedup or guide search become essential. In literature various techniques including parallelization [13], declaratively specifying graph extraction tasks over database schemas [14], and introduction of mode declarations [15] have been proposed to speed-up graph based concept discovery systems or reduce the search space.

In this study, we empirically evaluate performance of three bivariate statistical methods, namely *frequency ratio (FR)*, *hazard index (HI)*, and *weight of evidence (WoE)*, as heuristics to guide search in relational pathfinding-based concept discovery systems. All three methods report measure of relatedness between an outcome of an event and an effect that triggers the event. These methods have extensively been used for natural hazard modelling, particularly in landslide susceptibility assessment [16,17].

In context of relational pathfinding-based concept discovery, these statistical methods can be utilized to select the edge, i.e. relation, to extend candidate concept descriptors. To analyze performance of these bivariate statistical methods as heuristics, a relational pathfinding-based concept discovery system is implemented and run in four different settings. In the first setting, relations to extend candidate concept descriptors are chosen at random and in the remaining three settings relations are chosen according to their frequency ratio, hazard index, and weight of evidence values. Experiments conducted on three datasets show that all three methods provided speed up without loss in coverage when compared to randomly choosing relations to extend concept descriptors.

In this study, Neo4j graph database engine is used to store data and

¹ Department of Computer Engineering, Kocaeli University, Kocaeli, Turkey, ORCID ID: 0000-0003-0547-0653

² Department of Computer Engineering, Kocaeli University, Kocaeli, Turkey, ORCID ID: 0000-0002-6726-3679

³ Vadi Kurumsal Bilgi Sistemleri, Istanbul, Turkey, ORCID ID: 0000-0002-0798-2874

* Corresponding Author: Email: alev.mutlu@kocaeli.edu.tr

all data retrieval operations are implemented as Cypher queries – Neo4j’s graph query language.

Graph databases are database systems in which data schema and instances are modeled using graph structures or their extensions, i.e. hypernodes and hyperedges, and data manipulation operations are expressed as graph-oriented operations such as traversals [18]. Graph databases have recently become popular due to their flexible data representation and querying capabilities. Graph databases are of particular use when the concern is related to the relationship between the entities rather than the entities themselves [19]. As relational pathfinding-based concept discovery define concept descriptors by means of relations among entities, such database systems are ideal for data representation.

The rest of this paper is organized as follows. In Section 2 we introduce the concept discovery problem and summarize related work. In Section 3 we introduce frequency ratio, hazard index, and weight of evidence methods and their integration into the relational pathfinding-based concept discovery problem. In Section 4 we present and discuss the experimental results. The last section concludes the paper.

2. Concept Discovery

Given a set of positive and negative target instances that belong to a target relation and related background knowledge, concept discovery aims to find complete and consistent definitions of the target relation in terms of the background knowledge and possibly the target relation itself. More formally, assuming that E is the set of positive and negative target instances, $E = E^+ \cup E^-$, B is background knowledge; concept discovery problem can be formulated as inducing a set of hypothesis such that the following four conditions hold.

- **Prior satisfiability:** $B \wedge E^- \neq \square$
- **Posterior satisfiability:** $B \wedge H \wedge E^- \neq \square$
- **Prior Necessity:** $B \neq E^+$
- **Posterior sufficiency:** $B \wedge H \models E^+$

The sufficiency criterion is related to completeness, e.g. H should model all of the positive target instances relative to B , and posterior satisfiability is related to consistency, e.g. H should not model any of the negative target instances relative to B . However, due to noisy nature of real life data, completeness and consistency are extended to, respectively, explaining as many positive target instances as possible and as few negative target instances as possible.

Concept discovery systems can be classified into two groups: predictive systems and descriptive systems. Predictive concept discovery systems perform predictive induction to learn classification rules while descriptive systems perform descriptive induction to find regularities in data. Predictive systems input both positive and negative target instances and aim to induce complete and consistent hypothesis set while descriptive systems input only positive target instances and aim to induce maximally specific hypothesis set that explains all of the target instances.

Algorithm 1 outlines generic concept discovery process. It inputs a set of target instances and background knowledge and builds an initial set of concept descriptors. While termination conditions are not met, these concept descriptors are refined, evaluated and those that do not qualify user-defined metrics are pruned. Target instances modeled by concept descriptors are removed from the target instance set, the $Cover()$ method, and the process restarts with the remaining target instances. Termination conditions define restrictions on properties such as maximum rule length and maximum number of refinements; minimum number of target instances a concept descriptor should explain.

```

function GenericConceptDiscovery
Input: E, B: Target instances,
background knowledge
Output: solC: Concept descriptors
While E <> {}
    C = BuildInitialHypothesisSet()
    While termination criteria not met
        C = Refine(C)
        EvaluateAndPrune(C, E, B)
    End_while
    E = E - Cover(C, T)
    solC = solC + C
End_while

```

Algorithm 1: Generic Concept Discovery Process

The concept discovery problem has initially been studied within ILP research where relation data is represented within first order logic framework and candidate concept descriptors are refined using logical operators such as absorption operator of inverse entailment and least general generalization. Although ILP-based concept discovery systems have been successfully applied to several problems they suffer from scalability and efficiency issues [20] and are vulnerable to local plateau problem [11].

Recently the problem has been addressed from graph mining perspective. Graphs provide flexible means to represent relational data and have well studied algorithms that can be utilized for concept discovery. Graph-based approaches for concept discovery can be classified into two: substructure-based and relational pathfinding-based. The former focuses on substructures and assumes that substructures that satisfy some user defined criteria and include structures that represent positive target instances are concept descriptors. The later focuses on paths and assumes that frequent paths that include graph elements that represent target instances are concept descriptors.

Graph-based concept discovery systems generally follow hill-climbing strategy to build concept descriptors. However in case of large graphs such a setting becomes computationally inefficient. To speedup the graph-based concept discovery, methods such as simultaneous covering [12] and introduction of mode declarations [15] have been proposed.

More recently, studies that focus on mining graph databases have also been published. Such studies include crime analysis [21], searching semantically similar subgraphs [22], frequent itemset mining [23]. Applications of graph databases in concept discovery problem have been addressed in [24,25].

3. The Heuristics

In this section we firstly explain bivariate statistics in general and later provide definitions of those used in this study. In this section we also explain how these statistics are incorporated into relational pathfinding-based concept discovery process.

Bivariate statistics analyze empirical relationship between two variables, one of which is generally called independent variable and the other dependent variable. Bivariate statistics can be used to predict the value of the dependent variable if value of the independent variable is known. Types of bivariate statistical analysis include scatter plots that provide visual idea of the pattern that the dependent and the independent variables follow, regression analysis that provide equations of the patterns the variables follow, and correlation coefficients where the coefficients indicate degree of the relatedness between the variables.

Frequency ratio, hazard index and weight of evidence are bivariate statistical methods that provide correlation information between an event and a factor that triggers that event. Value 0 indicates no relatedness between the variable and the event, while positive values indicate positive causal relatedness.

Frequency ratio is formulated in (1), where i indicates value of a particular factor that effects the event, $\#P_i$ indicates the number of positive observation with value i , $\#P$ indicates total number of positive observations, $\#O_i$ indicates number of observations with value i , and $\#O$ indicates the total number of observations.

$$FR_i = \frac{\#P_i/\#P}{\#O_i/\#O} \quad (1)$$

Hazard index is defined in (2) where explanations of the parameters are the same of frequency ratio.

$$HI = \frac{\#P_i/\#O_i}{\#P/\#O} \quad (2)$$

Weight of evidence is a statistical method that uses log linear form of Bayesian probability model to estimate relative evidence. It is formulated in (3) where $\#P_i$ indicates presence of i value for a factor that triggers an event, $\#S$ and $\#S'$ indicate, respectively, positive and negative outcomes of the event. Nominator and dominator indicate conditional probabilities.

$$W_i^+ = \log_e \frac{P(\#P_i|\#S)}{P(\#P_i|\#S')} \quad (3)$$

In concept discovery there exists positive and negative target instances and their related background knowledge. To apply these statistical methods to the concept discovery problem, we assume that positive target instances correspond to positive outcomes of an event and negative target instances correspond to negative outcomes of an event. Relations in the background knowledge correspond to triggering factors of an event.

Algorithm 2 outlines the generic concept discovery process enhanced with the heuristics. The modified version of the algorithm inputs target instances, background knowledge and the setting it will run in. The algorithm starts with generating initial set

```

function HeuristicBased_ConceptDiscovery
Input: E, B, S: Target instances,
background knowledge, setting
Output: solC: Concept descriptors
While E <> {}
    C = BuildInitialHypothesisSet()
    While termination criteria not met
        For_each  $b_i$  in B
            If (S == 0)
                HV[i] = 0
            If (S == 1)
                HV[i] = CalculateFR( $b_i$ , E)
            If (S == 2)
                HV[i] = CalculateHI( $b_i$ , E)
            If (S == 3)
                HV[i] = CalculateWoE( $b_i$ , E)
        End_for
        C = Refine(C, max(HV))
        B = B - max(HV)
        EvaluateAndPrune(C, E, B)
    End_while
    E = E - Cover(C, T)
    solC = solC + C
End_while

```

Algorithm 2: Heuristic based concept discovery

of candidate concept descriptors. For each background relation, based on the setting parameter, the algorithm calculates FR, HI, or WoE values and populates an array with these values. The algorithm chooses the relation with the maximum value, using $max()$ method, and extends the current set of concept descriptors accordingly. The algorithm removes this relation from the background knowledge and performs the same actions until termination criteria are met. Once termination criteria are satisfied, target instances explained by the concept descriptors are removed from the target instance set and the process restarts. If the setting is 0 then the array is filled with 0s and $max()$ method returns a random relation.

4. Experimental Results

In order to evaluate the performance of FR, HI, and WoE as heuristics to guide the search in relational pathfinding-based concept discovery a set of experiments are conducted on three benchmark datasets. In this section we firstly introduce the datasets used in the experiments and the experimental setting and next we discuss the performance of the proposed heuristics in terms of search space reduction, running time, and accuracy.

4.1. Dataset Properties and Experimental Settings

In Table 1 we list the properties of the datasets used in evaluation of the proposed heuristics. Eastbound is a dataset describing properties of trains that travel either east or west and the problem is to find definitions of the trains that travel east. Mesh dataset is about finite element methods of engineering and the problem is to find definitions of edges. The last dataset, namely Student Loan, describes properties of students who are overdue or punctual in their loan payments and the problem is to find definitions of punctual students. The minimum support and confidence as well as maximum rule length parameters are set according to [26]. In Table 1 we also list properties of the graphs that correspond to the datasets. In Table 1 and the subsequent tables EB corresponds to the Eastbound dataset, M corresponds to the Mesh dataset, and SL corresponds to the Student Loan dataset.

Table 1. Dataset properties and experimental settings

	EB	M	SL
# Predicates	12	26	10
# Target Instances	30	76	1000
# Background Inst.	191	233	4288
Min. Support	0.1	0.1	0.1
Min. Confidence	0.1	0.7	0.7
Max Rule Length	3	3	3
# Vertices	49	107	1098
# Edges	183	308	6593

4.2. Analysis of Search Space Reduction and Running Time

In context of concept discovery, search space corresponds to the total number of candidate concept descriptors generated and evaluated during the concept discovery process.

In Table 2, we report the number of candidate concept descriptors generated in different settings. The $\#CD$ column indicates the number of relational paths generated and evaluated and the R column indicates the reduction of the search space, in percentage, with respect to the Random setting. The presented results are obtained by 10-fold experiments. As the results show for the Eastbound and Mesh datasets the proposed heuristics reduced the search space such that WoE performed the best. However, for the Student Loan dataset none of the heuristics provided any

improvement. This indeed is due to the fact that any randomly generated concept descriptor of length one or two can model enough number of positive target examples to satisfy the minimum support and limited number of negative target instances such that they do not violate the minimum confidence.

Table 2. Search space reduction

	Random		FR		HI		WoE	
	#CD	R	#CD	R	#CD	R	#CD	R
EB	70.7	57	19.37	57	19.37	56	20.79	
M	67.5	58	14.07	58	14.07	30	55.55	
SL	88.2	202	-	202	-	197	-	

In Table 3 we present the speedup results obtained. Speedup is measure to compare performance of different approaches in solving the same problem in terms of their running time. Speedup is calculated by dividing running of time of the original algorithm by the running time of the modified version of the algorithm. The RT columns indicate running time in seconds, and the S columns indicate speedups relative to the Random setting. A speedup value greater than 1 indicates improvement in running time. The reported running times in Table 3 are average of 10-fold experiments. As the experimental results show each heuristic provided improvement in running time. However there is no heuristic that provides the best speedup for all cases.

Table 3. Speed up

	Random		FR		HI		WoE	
	RT	S	RT	S	RT	S	RT	S
EB	22.01	9.81	2.24	9.88	2.22	10.58	2.08	
M	24.27	12.86	1.88	11.89	2.04	8.76	2.77	
SL	72.48	59.44	1.21	62.68	1.15	61.15	1.18	

Although for the Student Loan dataset, the Random setting generates less number of candidate concept descriptors it has a longer running time when compared to the other settings. This is due to the fact that average lengths of candidate concept descriptors generated by heuristic aided settings are shorter than those generated by the Random setting.

In Table 4 we report we report the average length of the candidate concept descriptors. As the table indicates, heuristic based settings generate shorter candidate concept descriptors when compared to the random setting.

Table 4. Average concept descriptor length

	Random	FR	HI	WoE
EB	1.75	1.4	1.4	1.2
M	1.73	1.6	1.6	1.3
SL	1.32	1.13	1.13	1.13

In order to statistically validate the running times of the four settings, we performed the Friedman and the Holm's post hoc tests. The Friedman test is a non-parametric test that finds if there exists difference among treatments across multiple attempts [27] The Holm's post-hoc test is used to find the differing groups if there are [27].

In Figure 1 we illustrate Holm's Post Hoc analysis considering all

four settings for the Eastbound dataset. Friedman test's p-value for all settings is below 0.05 hence the methods statistically differ by means of running time. Holm's test indicates that HI ve FR rank lower, i.e. have shorter running times, compared to the Random setting and WoE. When compared in pairs, Friedman test returned p-value = 2.2e-16 for Random and WoE indicating that WoE based setting has statistically shorter running time. Friedman test's p-value for HI and FR settings was 0.5554 indicating that two methods do not statistically differ.

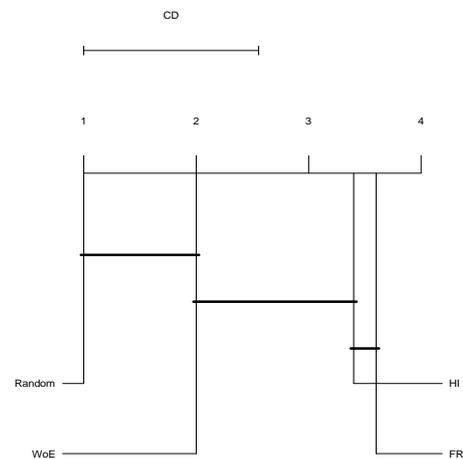


Fig. 1. Holm's Post hoc test for the Eastbound dataset (Friedman test p = 7.416e-14)

Friedman test's p-value considering the Mesh dataset for all the settings was < 2.2e-16, indicating that methods statistically differ. As Figure 2 depicts, HI and WoE have shorter running times compared to FI and the Random setting. When compared in pairs, FR is superior over the Random and WoE is superior over HI both with p-value = 2.2e-16.

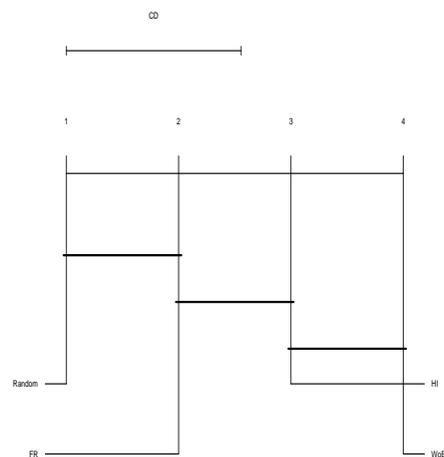


Fig. 2. Holm's Post hoc test for the Mesh dataset (Friedman test p-value = 7.416e-14)

Figure 3 shows Holm's Post Hoc test results for the Student Loan dataset considering all of the settings. As the figure shows, WoE and FR statistically differ from HI and the Random setting and have shorter running times. When compared in pairs, the Random setting and HI do not differ statistically (Friedman's test p-value = 0.55); while FR and WoE statistically differ (Friedman's test p-value=2.2e-16) such that FR is superior over WoE.

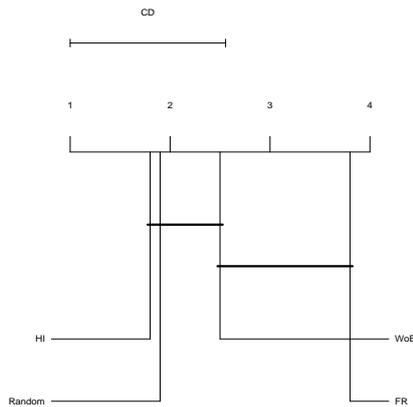


Fig. 3. Holm's Post Hoc test for the Student Loan dataset (Friedman test p-value = 0.0002176)

4.3. Analysis of Predictive Accuracy

In this section we discuss predictive accuracy results of the heuristics. For this purpose, we employ Receiver Operating Characteristics (ROC) curve analysis, which is a function of true positive rate (TPR) against false positive rate (FNR). In case of scoring classifiers, ROC analysis generates multiple points in the ROC space by considering all possible thresholds while it generates only one point and two line segments in case of binary crisp classifier [28].

TPR and FNR are formulated in (4) and (5), respectively, where TP indicates the number positive instances classified as positive and FN indicates the number of negative instances incorrectly classified as positive.

$$TPR = \frac{TP}{TP+FP} \quad (4)$$

$$FNR = \frac{FN}{FN+TP} \quad (5)$$

Predictive accuracy of a classifier can be measured by the area under the ROC curve (AUC). An area of 1 indicates a perfect classifier while an area of 0.5 indicates worthless classifier.

Figure 4 plots ROC curve and AUC values for the Eastbound dataset considering all four settings. As AUC values indicate, heuristic-based settings have better predictive accuracy compared to the random setting. As ROC curves for HI and FR overlap, only ROC curve of HI is visible on the plot.

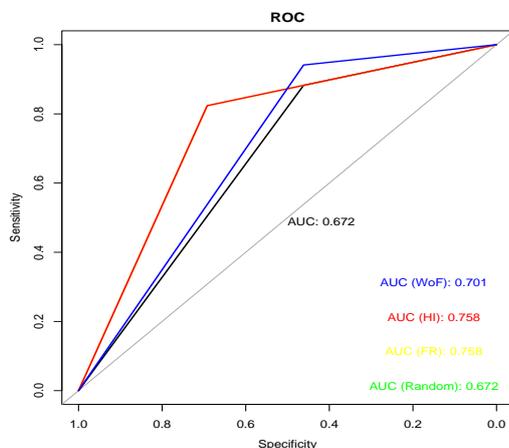


Fig. 4. ROC curve analysis for the Eastbound dataset

In Figure 5, we plot ROC curve and AUC values for the Student Loan dataset. As the AUC values show, the heuristic-based settings are superior over the Random setting by means of predictive accuracy.

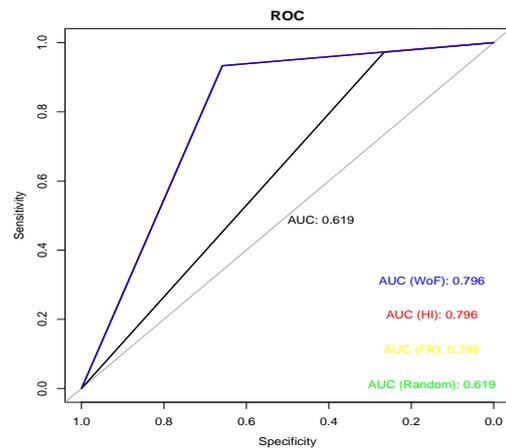


Fig. 5. ROC curve analysis for the Student Loan dataset

Mesh dataset belongs to descriptive learning task and rules define properties of different mesh values. Hence for this dataset we do not plot ROC curve but instead provide the number of explained target instances. All heuristic aided settings discovered concept descriptors that explain 28 of the target instances while concept descriptors discovered by the Random setting could explain 25 target instances.

5. Conclusion

This study provides experimental analysis of three bivariate statistical methods as heuristics to guide search in relational pathfinding-based concept discovery systems. Bivariate statistics provide degree of association between variables and in this study they are utilized to report relatedness of target instance and background knowledge. The experimental results conducted on three datasets showed that the studied bivariate statistical methods can be used as heuristics to guide search in relational pathfinding-based systems as they reduced the search space and provided speedup and increased predictive accuracy. The datasets used in this study included background knowledge that is directly related to target instances. As a future work we plan to utilize these bivariate statistical methods to provide relatedness information between target instances and their indirectly related background knowledge.

References

- [1] S. Dzeroski, "Multi-relational data mining: an introduction", *ACM SIGKDD Exploration Newsletter*, vol. 5, no. 1, pp. 1-16, 2003.
- [2] S. Muggleton and L. De Raedt, "Inductive logic programming: Theory and methods", *The Journal of Logic Programming*, vol. 19, pp. 629-679, 1994.
- [3] J. R. Quinlan, "Learning logical definitions from relations", *Machine Learning*, vol. 5, no. 3, pp. 239-266, 1990.
- [4] A. Srinivasan, "The aleph manual", 2001.
- [5] S. Muggleton, "Inverse entailment and PROGOL", *New Generation Computing*, vol. 13, no. 3-4, pp. 245-286, 1995.
- [6] M. J. Sternberg and S. Muggleton, "Structure activity relationships (SAR) pharmacophore discovery using inductive logic programming (ILP)", *QSAR and Combinatorial Science*, vol. 22, no. 5, pp. 527-532, 2003.
- [7] H. Blockeel, S. Dzeroski and J. Grbovic, "Simultaneous prediction of

- multiple chemical parameters of river water quality with TILDE”, in *Proc. European Conference of Data Mining and Knowledge Discovery*, Prague, Czech Republic, 1990, pp. 32-40.
- [8] B. Dolsak, I. Bratko and A. Jezernik, “Knowledge base for finite-element mesh design learned by inductive logic programming”, *AI EDAM*, vol. 12, no. 2, pp. 95-106, 1998.
- [9] D. J. Cook, L. B. Holder and S. Djoko, “Knowledge discovery from structured data”, *Journal of Intelligent Information Systems*, vol. 5, no. 3, pp. 229-248, 1995.
- [10] J. A. Gonzalez, L. B. Holder and D. J. Cook, “Graph-based concept learning”, in *Proc. FLAIRS Conference*, Key West, Florida, USA, 2001, pp. 377-381.
- [11] B. L. Richards and R. J. Mooney, “Learning relations by pathfinding”, in *Proc. National Conference on Artificial Intelligence*, San Jose, CA, USA, 1992, pp. 50-55.
- [12] Z. Gao, Z. Zhang and Z. Huang, “Learning relations by pathfinding and simultaneous covering”, in *Proc. World Conference on Computer Science and Information Engineering*, Los Angeles, CA, USA, 2009, pp. 539-543.
- [13] D. J. Cook, L. B. Holder, G. Galal and R. Maglothin, “Approaches to parallel graph-based knowledge discovery”, *Journal of Parallel and Distributed Computing*, vol. 61, no. 3, pp. 427-446, 2001.
- [14] K. Xirogiannopoulos and A. Deshpande, “Extracting and analyzing hidden graphs from relational databases”, in *Proc. ACM International Conference of Management of Data*, Chicago, IL, USA, 2017, pp. 897-912.
- [15] I. M. Ong, I. de Castro Dutra, D. Page and V. S. Costa, “Mode directed path finding”, in *Proc. European Conference on Machine Learning*, Porto, Portugal, 2005, pp. 673-681.
- [16] S. Lee and B. Pradhan, “Landslide hazard mapping at Selangor, Malaysia using frequency ratio and logistic regression models”, *Landslides*, vol. 4, no. 1, pp. 33-41, 2007.
- [17] N. R. Regmi, J. R. Giardino and J. D. Vitek, “Modeling susceptibility to landslides using weight of evidence approach: Western Colorado, USA”, *Geomorphology*, vol. 115, no. 1-2, pp. 172-187, 2010.
- [18] R. Angles and C. Gutierrez, “A survey of graph database models”, *ACM Computing Surveys*, vol. 40, no. 1, pp. 1-39, 2008.
- [19] A. Silvescu, D. Caragea and A. Atramentov, “Graph Databases”, Artificial Intelligence Research Laboratory Department of Computer Science, Iowa State University, 2012, [online] <http://people.cs.ksu.edu/~dcaragea/papers/report.pdf>.
- [20] Q. Zeng, J. M. Patel and D. Page, “QuickFOIL: scalable inductive logic programming”, *Proceedings of the VLDB Endowment*, vol. 8, no. 3, pp. 197-208, 2014.
- [21] S. Sathyadevan, M. S. Devan and S. Gangadharan, “Crime analysis and prediction using data mining”, in *Proc. IEEE First International Conference on Networks and Soft Computing*, 2014, pp. 406-412.
- [22] X. Yu, Y. Sun, P. Zhao, J. Han, “Query driven discovery of semantically similar substructures in heterogeneous networks”, in *Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 1500-1503.
- [23] M. Jayashree and M. Barathi, “Parallel Mining of Frequent Itemsets using FINM on Neo4j”, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, 2017.
- [24] F. Goz and A. Mutlu, “Learning Logical Definitions of n-ary Relations in Graph Databases”, in *Proc. The 13th International Conference on Hybrid Artificial Systems*, Ovideo, Spain, 2018, pp. 50-61.
- [25] N. C. Abay, A. Mutlu and P. Karagoz, “A Graph-based Concept Discovery Method for n-ary Relations”, in *Proc. The 17th International Conference on Big Data Analytics and Knowledge Discovery*, Valencia, Spain, 2015, pp. 391-402.
- [26] A. Mutlu and P. Karagoz, “Policy-based memoization for ilp-based concept discovery systems”, *Journal of Intelligent Information Systems*, vol. 46, no. 1, pp. 99-120, 2016.
- [27] J. Derrac, S. Garcia, D. Molina and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms”, *Swarm and Evolutionary Computing*, vol. 1, no. 1, pp. 3-18, 2011.
- [28] R. C. Prati, G. E. A. P. A. Batista and M. C. Monard, “A survey on graphical methods for classification predictive performance evaluation”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1601-1618, 2011.