Cosine Similarity-Based Pruning for Concept Discovery

Abdullah Dogan[1](✉), Alev Mutlu[2], and Pinar Karagoz[1]

[1] Department of Computer Engineering,
Middle East Technical University, Ankara, Turkey
{adogan,karagoz}@ceng.metu.edu.tr
[2] Department of Computer Engineering, Kocaeli University, Kocaeli, Turkey
alev.mutlu@kocaeli.edu.tr

Abstract. In this work we focus on improving the time efficiency of Inductive Logic Programming (ILP)-based concept discovery systems. Such systems have scalability issues mainly due to the evaluation of large search spaces. Evaluation of the search space cosists translating candidate concept descriptor into SQL queries, which involve a number of equijoins on several tables, and running them against the dataset. We aim to improve time efficiency of such systems by reducing the number of queries executed on a DBMS. To this aim, we utilize cosine similarity to measure the similarity of arguments that go through equijoins and prune those with 0 similarity. The proposed method is implemented as an extension to an existing ILP-based concept discovery system called Tabular Cris w-EF and experimental results show that the poposed method reduces the number of queries executed around 15 %.

1 Introduction

Concept discovery [3] is a multi-relational data mining task and is concerned with inducing logical definitions of a relation, called *target relation*, in terms of other provided relations, called *background knowledge*. It has extensively been studied under Inductive Logic Programming (ILP) [12] research and successful applications are reported [2,4,7,10].

ILP-based concept discovery systems consist of two main steps, namely *search space formation* and *search space evaluation*. In the first step candidate concept descriptors are generated and in the second step candiate condept descriptors are converted into queries, i.e. SQL queries, and are run against the dataset. As the search space is generally large and the queries involve multiple joins over several tables, the second step is computationally expensive and dominates the total running time of a concept discovery system. Several methods such as parallelization, memoization have been investigated to improve running time of the search space evaluation step.

In this paper we propose a method that improves the running time of concept discovery systems by reducing the number of SQL queries run on a database. The proposed method calculates the cosine similarity of the tables that appear

T. Czachórski et al. (Eds.): ISCIS 2016, CCIS 659, pp. 90–96, 2016.
DOI: 10.1007/978-3-319-47217-1_10

in a query, and prunes those with 0 similarity. To realize this, (i) term-document count matrix where domain values of arguments of tables correspond to terms and relation arguments correspond to documents is built, and (ii) cosine similarity of table arguments that participate in a query are calculated from the term-document count matrix and those with 0 similarity are pruned.

The proposed method is implemented as an extension to an existing concept discovery system called Tabular CRIS w-EF [14,15]. To evaluate the performance of the proposed method several experiments are conducted on data sets that belong to different learning problems. The experimental results show that the proposed method reduces the number of queries executed by 15 % on the average without any loss in the accuracy of the systems.

The rest of the paper is organized as follows. In Sect. 2 we provide the background related to the study, in Sect. 3 we introduce the proposed method, and in Sect. 4 we present and discuss the experimental results. Last section concludes the paper.

2 Background

Concept discovery is a predictive multi relational data mining problem. Given a set facts, called *target instances*, and related observations, called *background knowledge*, concept discovery is concerned with inducing logical definitions of the target instances in terms of background knowledge. The problem has primarily been studied by ILP community and successful application have been reported.

In ILP-based concept discovery systems data is represented within first order logic framework and concept descriptors are generated by specialization or generalization of some an initial hypothesis. ILP-based concept discovery systems follow generate and test approach to find a solution and usually build large search spaces. Evaluation of the search space consists of translating concept descriptors into queries and running them against the data set. Evaluation of the queries is computationally expensive as queries involve multiple joins over tables. To improve running time of such systems several methods including parallelization [9], caching [13], query optimization [20] have been proposed. In parallelization based approaches either the search space is built or evaluated in parallel by multiple processors, in caching based methods queries and their results are stored in hash tables in case the same query is regenerated, and in query optimization based approaches several query optimization techniques are implemented to improve the running time of the search space evaluation step.

Cosine similarity is a popular metric to measure the similarity of data that can be represented as vectors. Cosine similarity of two vectors is the inner product of these vectors divided by the product of their lengths. Cosine similarity of -1 indicates exactly opposition, 1 indicates exact correlation, and 0 indicates decorrelation between the vectors. It has been applied in several domains including text document clustering [5], face verification [16].

In this work we propose to measure the cosine similarity of table arguments that partake in equijoins and prune those with cosine similarity of 0 without running them against the data set. To achieve this, firstly we group attributes that belong to the same domain, build a term-document matrix for each domain where domain values of the attributes constitute the terms, and individual arguments constitute the documents. When two arguments go through an equijoin we calculate their cosine similarity from the term-document matrix and prune those queries that have cosine similarity of 0. The proposed method is implemented as an extension to an existing ILP-based concept discovery system called Tabular CRIS w-EF. Tabular CRIS w-EF is an ILP-based concept discovery system that employs association rule mining techniques to find frequent and strong concept descriptors and utilizes memoization techniques to improve search space evaluation step of its predecessor CRIS [6].

3 Proposed Method

ILP-based systems represent the concept descriptors as Horn clauses where the positive literal represents the target relation, and the negated literals represent relations from the background knowledge. To evaluate such clauses, they are translated into SQL queries, where relations constitute the FROM clause and argument values form the WHERE clause of the query. As an example, consider the concept descriptor like *brother(A, B):-mother(C, A), mother(C, B).* This concept descirptor is mapped to the following SQL query:

```
SELECT SELECT b.arg1, brother.arg2
FROM brother AS b, mother AS m1, mother AS m2
WHERE brother.arg1=m1.arg2 and b.arg2=m2.arg2 and m1.arg1=m2.arg1
```

Fig. 1. Sample concept descriptor evaluation query

In such a transformation argument values with the same value go through equijoins. The proposed method targets such equijoins and prevents execution of queries that involve equjoins whose participating arguments have cosine similarity 0.

To achieve this,

(1) arguments are grouped based on their domains,
(2) for each such group term-document matrix is formed where values of the domain are the terms, arguments are the documents and values of an argument is the bag of the words of the argument
(3) for each term-document matrix a cosine similarity matrix is calculated.

To populate the count vector of an argument of a relation, i.e. *rel(arg1, . . . , argn)* the following SQL statement is executed

ILP-based concept discovery systems construct concept descriptors in an iterative manner. At each iteration, a concept descriptor is specialized by appending

```
SELECT arg1, COUNT(*)-1 vector FROM
        (SELECT arg1 FROM rel
                UNION ALL
        SELECT arg1 FROM rel_domain) t
GROUP BY arg1;
```

Fig. 2. Query for creating a count vector for rel.arg1

a new literal to the body of the concept descriptor in order to reduce the number of negative target instances it models, and it is evaluated. The proposed method inputs the refined concept descriptors, and checks if the newly added literal causes an equijoin. If and equijoin is detected, the cosine similarity of the arguments is fetched from the previously built matrix. If the cosine similarity is 0 then the concept descriptor is pruned, otherwise it is evaluated against the data set. If the newly added literal does not produce an equijoin then the query is directly evaluated against the data set. The proposed method is outlined in Algorithm 1.

Algorithm 1. PruneBasedOnSimilarity(vector<conceptDescriptors> C)

```
for (i = 0; i < C.size() ; i++) do
  newLiteral=C[C[i].literals.size()]
  for (j = 0; j < C[i].literals.size() - 1; j++) do
    for (k = 0; k < C[i].literals[j].arguments.size(); k++) do
      for (m = 0; m < newLiteral.argument.size(); m++) do
        if (C[i].literals[j].argument[k]=newLiteral.argument[m] AND similarity(C[i].literals[j].argument[k],newLiteral.argument[m])==0) then
          prune pC[i]
        end if
      end for
    end for
  end for
end for
```

In literature, there exists several ILP-based concept discovery systems that work on Prolog engines [11,17]. Such systems benefit from depth bounded interpreters for theorem proving to test possible concept descriptors. The proposed method is also applicable for such systems, as in Prolog notation each predicate can be considered a table and arguments of the literal can be considered as the fields of the table. With such a transformation, the proposed method can be utilized to prune hypotheses for ILP-based concept discovery systems that work on Prolog like environments.

In terms of algorithmic complexity, the proposed method consists of two main steps (i) matrix construction and (ii) cosine similarity calculation. To construct the matrix, one SQL query needs to be run for each literal argument. Complexity of cosine similarity is quadratic, hence applicable to real world data sets.

4 Experimental Results

To evaluate the performance of the proposed method we conducted experiments on data sets with different characteristics. Table 1 lists the data sets used in the experiments. Dunur and Elti are family relationship datasets. They are Turkish terms and are defined as follows: A is dunur of B if a child of A is married to a child of B, A is elti of B if As husband is brother of Bs husband. All the arguments of the two data sets belong to the same domain and both data sets are highly relational. Mutagenesis [19] and PTE [18] are biochemical datasets and aim is to classify the chemicals as to being related to mutagenicity and carcinogenicity or not, respectively. Mesh [1] is an engineering problem dataset where the problem is to find rules that define mesh resolution values of edges of physical structures. In the Eastbound [8] dataset there are two types of trains: (a) those that travel east called eastbound; and those that travel west called westbound. The problem is to find concept descriptors that define properties of the trains that travel to east. In these data sets there several domains that arguments belong to. The experiments are conducted on MySQL version 5.5.44-0ubuntu0.14.04.1. The DBMS resides on a machine with Core i7-2600K CPU processor and 7.8 GB RAM.

Table 1. Experimental parameters for each used data sets

Data set	Num of relations	Num of instances	Argument types
Dunur	9	234	Categorical
Elti	9	234	Categorical
Eastbound	12	196	Categorical, real
Mesh	26	1749	Categorical, real
Mutagenesis	8	16,544	Categorical, real
PTE	32	29,267	Categorical, real

In Table 2 we report the experimental results. Filtering Queries column shows the decrease in the number of queries when the proposed method is employed. The experimental results show that the proposed method performs well on the data sets that are highly relational, i.e. Dunur and Elti data sets. The proposed method performs sligly worse for the data sets that contains numerical attributes as well as categorical attributes to theose that only contains categorical attributes. This is indeed due to the fact that, arguments from the categorical domain go through equijoins, while arguments that belong to numerical domain go through less than (<), greater than (>) comparisons in SQL statements.

The last column of Table 2 reports the time impreovement when the proposed method is employed. When compared to decrease in the number of queries executed, the decrease in running time is less. This is due to the fact that Tabular CRIS w-EF employs advanced memoization mechanisms to store evaluation

Table 2. Improvements of proposed method

Data set	Tabular CRIS-wEF			Pruning by the proposed method			Improvement %		
	Num. rules	Num. queries	Time (mm:ss.sss)	Num. rules	Num. queries	Time (mm:ss.sss)	Rules	Queries	Time
Dunur	1887	5807	00:02.086	1279	4607	00:01.783	32.22	20.66	14.54
Elti	1741	5333	00:02.655	1422	4922	00:02.470	18.32	7.71	6.99
Eastbound	7294	34654	00:04.091	6805	32665	00:03.895	6.70	5.74	4.77
Mesh	56512	249084	00:27.302	54314	238982	00:27.314	3.89	4.06	−0.05
Mutagenesis	62486	223644	34:04.099	55477	216635	33:42.752	11.22	3.13	1.04
PTE	64322	237082	35:50.340	58503	231191	35:15.975	9.05	2.48	1.60
PTE No Aggr.	11166	43862	03:46.457	10328	43024	03:40.578	7.50	1.91	2.60

queries and retrieve results of repeated queries from hash tables. Nevertheless, the proposed method improves the running time of Tabular CRIS w-EF around 7.5 % on average.

5 Conclusion

Concept discovery systems face scalability issues due to the evaluation of the large search spaces they build. In this paper we propose a pruning mechanism based on cosine similarity to improve running time of concept discovery systems. The proposed method calculates the cosine similarity of arguments that participate in equijoins and prunes those concept descriptors that have arguments with cosine similarity 0. The proposed method is applicable to concept descovery systems that work on relational databases or Prolog like engines. The experimental results show that the proposed method decreased the number of concept descriptor evaluations around 15 % on the average, and improved the running time of the system around 7.5 % on the average.

References

1. Dolšak, B.: Finite element mesh design expert system. Knowl. Based Syst. **15**(5), 315–322 (2002)

2. Dolsak, B., Muggleton, S.: The application of inductive logic programming to finite element mesh design. In: Inductive Logic Programming, pp. 453–472. Academic Press (1992)
3. Dzeroski, S.: Multi-relational data mining: an introduction. SIGKDD Explor. **5**(1), 1–16 (2003). doi:10.1145/959242.959245
4. Feng, C.: Inducing temporal fault diagnostic rules from a qualitative model. In: Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA, pp. 403–406 (1991)
5. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, pp. 49–56 (2008)
6. Kavurucu, Y., Senkul, P., Toroslu, I.H.: ILP-based concept discovery in multi-relational data mining. Expert Syst. Appl. **36**(9), 11418–11428 (2009). doi:10.1016/j.eswa.2009.02.100
7. King, R.D., Muggleton, S., Lewis, R.A., Sternberg, M.: Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. Proc. Nat. Acad. Sci. **89**(23), 11322–11326 (1992)
8. Larson, J., Michalski, R.S.: Inductive inference of VL decision rules. ACM SIGART Bull. **63**, 38–44 (1977)
9. Matsui, T., Inuzuka, N., Seki, H., Itoh, H.: Comparison of three parallel implementations of an induction algorithm. In: 8th International Parallel Computing Workshop, pp. 181–188. Citeseer (1998)
10. Muggleton, S., King, R., Sternberg, M.: Predicting protein secondary structure using inductive logic programming. Protein Eng. **5**(7), 647–657 (1992)
11. Muggleton, S.: Inverse entailment and progol. New Gener. Comput. **13**(3–4), 245–286 (1995)
12. Muggleton, S., Raedt, L.D.: Inductive logic programming: theory and methods. J. Log. Program. **19**(20), 629–679 (1994). doi:10.1016/0743-1066(94)90035-3
13. Mutlu, A., Karagoz, P.: Policy-based memoization for ILP-based concept discovery systems. J. Intell. Inf. Syst. **46**(1), 99–120 (2016). doi:10.1007/s10844-015-0354-7
14. Mutlu, A., Senkul, P.: Improving hash table hit ratio of an ILP-based concept discovery system with memoization capabilities. In: Gelenbe, E., Lent, R. (eds.) Computer and Information Sciences III, pp. 261–269. Springer, London (2012). doi:10.1007/978-1-4471-4594-3_27
15. Mutlu, A., Senkul, P.: Improving hit ratio of ILP-based concept discovery system with memoization. Comput. J. **57**(1), 138–153 (2014). doi:10.1093/comjnl/bxs163
16. Nguyen, H.V., Bai, L.: Cosine similarity metric learning for face verification. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6493, pp. 709–720. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19309-5_55
17. Quinlan, J.R.: Learning logical definitions from relations. Mach. Learn. **5**(3), 239–266 (1990)
18. Srinivasan, A., King, R.D., Muggleton, S.H., Sternberg, M.J.: The predictive toxicology evaluation challenge. In: IJCAI, vol. 1, pp. 4–9. Citeseer (1997)
19. Srinivasan, A., Muggleton, S.H., Sternberg, M.J., King, R.D.: Theories for mutagenicity: a study in first-order and feature-based induction. Artif. Intell. **85**(1), 277–299 (1996)
20. Struyf, J., Blockeel, H.: Query optimization in inductive logic programming by reordering literals. In: Horváth, T., Yamamoto, A. (eds.) ILP 2003. LNCS (LNAI), vol. 2835, pp. 329–346. Springer, Heidelberg (2003). doi:10.1007/978-3-540-39917-9_22