Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

Using Subgroup Discovery Metrics to Mine Interesting Subgraphs

Russ Neely
Department of Computer Science
Tennessee Technological University
Cookeville, TN USA
rjneely42@students.tntech.edu

Zach Cleghern
Department of Computer Science
Tennessee Technological University
Cookeville, TN USA
zbcleghern42@students.tntech.edu

Douglas A. Talbert
Department of Computer Science
Tennessee Technological University
Cookeville, TN USA
dtalbert@tntech.edu

Abstract

While extensive work has been done in both graph mining and subgroup discovery, the potential benefits of combining the two fields have not been well studied. We propose, implement, and evaluate an adaption of an existing subgroup discovery algorithm to mine graph data. Our experiments use two different metrics from the subgroup discovery literature to demonstrate value in using such metrics to guide subgraph discovery and to build a foundation to support further studies combining subgroup discovery and graph mining.

Introduction

The structural relationships revealed in graph representations of data enables discovery of knowledge that may not be as easily mined from other data representations. As part of our long-term goal of using knowledge discovery to improve healthcare delivery, we seek to apply graph mining to healthcare utilization data.

Specifically, we want to discover graph-based patterns that are associated with high-quality/low-cost care. Subgraph discovery, however, typically searches for dominant (e.g., most frequent or largest) patterns. We desired a richer palette of heuristics to guide subgraph identification.

The *subgroup discovery* literature presents numerous heuristics for discovering interesting patterns in data (Herrara, *et al* 2011). This paper seeks to demonstrate the utility of applying heuristics from subgroup discovery to subgraph mining by adapting an existing subgroup discovery algorithm to search for subgraphs instead of subgroups. We do not claim that this is the best approach, but instead, we use it to illustrate value in using ideas from subgroup discovery to guide the identification of subgraphs.

Subgroup discovery seeks *interesting* subgroups. Numerous metrics have been used to quantify the interestingness of discovered subgroups. This exploratory study compares subgraph discovery using two specific subgroup interestingness metrics with an existing subgraph discovery algorithm, SUBDUE (Cook, Holder 2000).

The next section presents relevant background information about subgroup discovery, subgraph mining, and SUBDUE. Then, we describe our experimental methodology, including the data used in our analysis along with our algorithm modifications. After that, we present our results followed by conclusions and ideas for future work.

Background

Subgroup Discovery

Given a population of objects and a property of those objects in which we are interested, subgroup discovery seeks to "discover the subgroups of the population that are statistically 'most interesting,' i.e. are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest." (Wrobel 2001).

The following rule illustrates a traditional subgroup definition:

$$(Att_i=Val_{i,j}) \wedge (Att_k<Val_{k,l}) \Rightarrow (Class=TRUE),$$

where the antecedent is a conjunction of attribute/value pairs and the consequent is membership in the target class.

Subgroup discovery is an example of *supervised descriptive induction* tasks that lie between and share goals with both predictive and descriptive induction (Novak, Lavrač, and Webb 2009).

The basic subgroup discovery algorithm that we use, SD, employs a heuristic, fixed-width beam search based rules interestingness and complexity to select subgroup definitions (Gamberger and Lavrač 2002).

Herrera, *et al.* (2011) categorize and define numerous quality metrics that have been used to guide the search for subgroups. The two metrics that we explore are the precision measure Q_g and *Unusualness*.

The precision measure Q_g is a ratio of the number of true positives to false positives with an additional generalization parameter g (Herrara, *et al.* 2011). The equation for Q_g is as follows:

$$Q_g(R) = TP/(FP + g),$$

where R is the rule being evaluated, TP is the number of true positives, FP is the number of false positives, and g is the generalization parameter.

Unusualness is the product of *coverage* and *accuracy gain*. (Lavrač, Flach, Zupan 1999). *Coverage* is the percentage of data items that satisfy the antecedent of the rule, and *accuracy gain* is the *positive predictive value* of the rule (TP/(TP+FP)) minus the percentage of total items that are in the target class. *Unusualness* is computed as follows:

$$Unusual(R) = P(cond) * (PPV(R) - P(TargetClass))$$

where *P(cond)* is the probability of an instance satisfying the conditions (antecedent) of a rule or the *coverage* of the rule, *PPV(R)* is the positive predictive value of the rule, *P(TargetClass)* is the probability of the target class.

Frequent Subgraph Mining & SUBDUE

Frequent subgraph mining is "the process of finding those subgraphs from a given graph or a set of graphs which have frequent or multiple instances within the given graph or the set of graphs" (Krishna, Ranga Suri, Athithan 2011).

SUBDUE is one such algorithm (Cook, Holder 2000). It can identify subgraphs by measuring the compression achieved when subgraph instances are replaced by a single representative vertex. In supervised mode, SUBDUE seeks to maximize compression in the target class graphs while minimizing compression in graphs outside the target class. In unsupervised mode, SUBDUE searches for normative subgraph patterns.

Methodology

Overview

We completed the following steps, described in detail below:

1. generate synthetic sets of supervised graphs with known subgraph distributions
2. modify an implementation of the SD subgroup discovery algorithm to search for subgraphs (called SD′) using the metrics defined above
3. apply SUBDUE and SD′ to the generated data sets to identify subgraphs
4. analyze results to evaluate the utility of SD′ and its subgroup discovery-based quality metrics to identify subgraphs

Data Generation

We generated data to simulate healthcare utilization data with procedure codes linked to the diagnostic codes justifying the procedure. The codes are parts of hierarchies that aggregate related codes into code classes (Figures 1 and 2). Each child node in these hierarchies is an instance of the parent node. In our data, only the leaves in the hierarchies represent actual procedures or diagnoses.

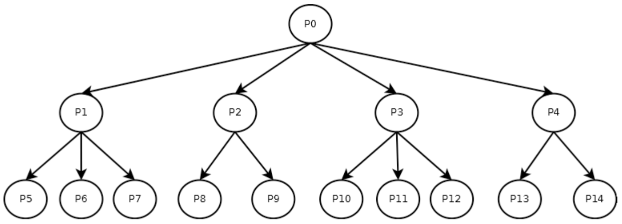

Figure 1. Synthetic procedure code hierarchy

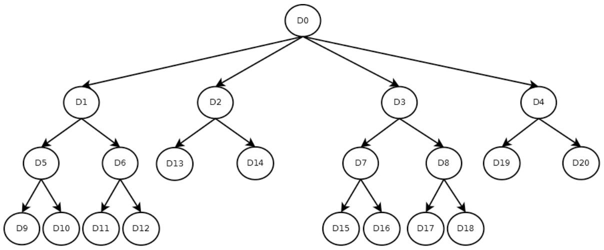

Figure 2. Synthetic diagnosis code hierarchy

Each graph represents a simulated patient's medical claims with vertices for procedures and diagnoses and an edge connecting each procedure to its associated diagnosis.

If a claim contains a procedure P5 is linked to a diagnosis D10, we show this with an edge from vertex P5 to vertex D10. Because of the "is-a" relationship in the code hierarchies, this link implies that there are also edges from P1 (P5's parent) to D10, from P5 to D5 (D10's parent), from P1 to D5, and so on, up to the roots of the hierarchy trees. Imputing these additional links in this domain allows discovered subgraphs to contain links between procedure classes and diagnosis classes. We believe this capability is very important for our medical domain.

We generated three sets, each with 10,000 patients (graphs). Each graph was assigned to either class 0 or class 1 (our target class). Table 1 describes the distribution of graphs across the two classes for each data set.

Table 1. Class distribution

Data set name	Class 0	Class 1
Balanced	50%	50%
Imbalanced_10	90%	10%
Imbalanced_5	95%	5%

Each patient was defined by the inclusion or exclusion of instances of a pair of subgraphs that we refer to as Subgraph 1 (Figure 3) and Subgraph 2 (Figure 4).

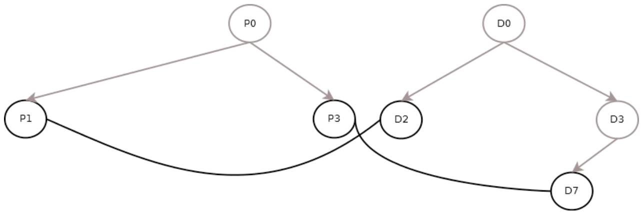

Figure 3. Subgraph 1 (relevant hierarchy graph is in gray)

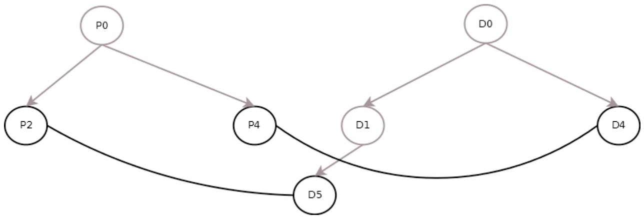

Figure 4. Subgraph 2 (relevant hierarchy graph is in gray)

The relative distribution of Subgraphs 1 and 2 in class 0 and class 1 was the same in all three data sets and is described in Table 2. The appearance of each subgraph is conditionally independent. These percentages were selected to represent different conditions that might influence subgraph discovery.

Table 2. Subgraph distribution for each class

Class	Subgraph 1	Subgraph 2
0	50%	1%
1	100%	50%

Each graph was generated by determining the class to which it belongs and which subgraph(s) (if any) it would include. Constituent subgraphs were included along with all the implied links involving parent nodes as described above. To add an element of randomness to the data, each patient graph has a 50% change of having an additional random claim (procedure/edge link).

The final step was to convert it into attribute-value pairs that the subgroup discovery algorithm could process. This was possible in our data domain because we could list each possible edge as an attribute and set its value to 1 if it was present in the graph and 0 if it was missing.

Modification of SD

Next, we modified the SD algorithm (Gamberger, Lavrač 2002) to search for subgraphs. We chose SD as our subgroup discovery algorithm for its simplicity, since we intend to show *value* in subgroup discovery metrics in subgraph discovery, not the *ideal* method. We started with the implementation of SD in the Orange data mining platform (Demšar, Curk, Erjavec 2013). To ensure that discovered subgraph definitions would not include conditions that excluded edges (where $edge_i$=0), our modified search algorithm, SD´, only considers edges that exist in the graphs. SD´ could be guided by either the precision measure Q_g or *Unusualness*.

Identification of Subgraphs

To evaluate the utility of subgroup discovery metrics in identifying subgraphs, we ran each of the three data sets (Balanced, Imbalanced_10, and Imbalanced_5) through SUBDUE and SD´. We used each of SUBDUE's three metrics (minimum description length, size, and set coverage) and both metrics in SD´ (Q_g and Unusualness). Our experiments used two different values for the parameter *g* in Q_g (10 and 100). Thus, we ran 18 different combinations of data sets and algorithms/metrics. For SUBDUE, we set the iteration parameter to 10, and for SD´, we set the minimum support to 1%.

Results

Our goal was to see if Subgraphs 1 and 2 could be found in each data set. Ideally, we would like to find each subgraph individually and precisely as it was defined.

Results with the 'Balanced' Dataset

Q_g with g = 10 on Balanced, we observed that the only significant subgraph it found was a more specific version of Subgraph 2. Q_g with g = 100, we were able to find Subgraph 2 as it was defined along with some unrelated edges. We were again unable to find Subgraph 1, only generalizations that were always accompanied by Subgraph 2 or its generalizations.

Applying *Unusualness* to Balanced, we observed the most general results, only producing variations of Subgraph 1 and 2 with generalized edges (for example, P4-D0 instead of P4-D4). Thus, we concluded it to be the *worst* of the three in finding subgraphs in Balanced.

SUBDUE on the Balanced set using Minimum Description Length (*MDL*) produced subgraphs that only contain *both* subgraphs as defined. Similarly, in SUBDUE with the *Size* option, we found that the best substructure contained both subgraphs as defined. SUBDUE with *Set Cover* on the Balanced set looked similar to the output of SD´. The initial (and most accurate) subgraph is a generalized version of Subgraphs 1 and 2.

Results with the 'Imbalanced_10' Dataset

Q_g (g = 10) was able to find one overly specific version of Subgraph 2 with no other edges, generalized results of both Subgraphs 1 and 2, and Subgraph 1 with an extra edge.

Q_g with g = 100 was less insightful. In each induced rule, an edge with P2 or D5 (or their parent vertices) is present. We were able to find another specific instance of Subgraph 2 and only generalizations of Subgraphs 1 and 2 together.

Unusualness showed the most promising results for this data set, finding Subgraph 1 as defined and Subgraph 2 as defined, respectively, with minimal excess edges. It also found generalizations of the subgraphs combined. This is the *only* metric and data set in which both Subgraph 1 and 2 were found as defined.

SUBDUE with *MDL* again generates the graph representing Subgraphs 1 and 2 as defined along with a pattern consisting of vertices and edges all around Subgraph 1, but not the hierarchy or the entire subgraph itself, but it is not able to find either subgraph by itself. The *Set Cover* option of SUBDUE finds a generic version of both Subgraph 1 and 2 similar to its performance on the Balanced set.

Results with the 'Imbalanced_5' Dataset

Q_g and g = 10, finds generalizations of both Subgraphs 1 and 2 and combinations of the two. We are not able to find Subgraph 1 or Subgraph 2 as defined. Q_g with g = 100, finds overly specific examples of Subgraphs 1 and 2 but not the combination of the two. *Unusualness* finds Subgraph 2 as defined with some extraneous graphs, variations of Subgraph 1, and generalizations of a combination of the two. SUBDUE with *MDL* (and with *Size*) found a subgraph combining Subgraph 1 and 2.

Discussion

No method could effectively discover clean versions of either of the two subgraphs by themselves. *Unusualness* on Imbalanced_10 came the closest. SUBDUE, however, did a good job overall in finding both patterns together.

Q_g with g = 10 did a reasonable job throughout in finding variants of Subgraph 2. With g = 100, Q_g provided more general results, sometimes finding Subgraph 1 or 2 as defined with some extra edges. Table 3 summarizes the results, indicating subgraphs found. The '+' indicates that the patterns were found together in the same subgraph.

Table 3. Summary of results

Method	Balanced	Imbalanced_10	Imbalanced_5
SD′ Q_g g = 10	SG2	SG2; SG2; SG1 + SG2	SG1; SG2; SG1 + SG2
SD′ Q_g g = 100	SG2	SG2; SG1 + SG2	SG1; SG2
SD′ *Unusualness*	SG1; SG2	SG1; SG2; SG1 + SG2	SG1; SG2; SG1 + SG2
SUBDUE *MDL*	SG1 + SG2	SG1 + SG2	SG1 + SG2
SUBDUE *Size*	SG1 + SG2	SG1 + SG2	SG1 + SG2
SUBDUE *SetCoverage*	SG1 + SG2	SG1 + SG2	SG1 + SG2

Conclusions and Limitations

Through our experiments, we have shown that metrics from subgroup discovery are capable of adding value to the subgraph discovery task by finding subgraphs that were different from what SUBDUE identified. In particular, we showed that there exist conditions where the *Unusualness* metric was able to isolate and discover known subgraphs better than any other technique we tested.

A limitation of SD′ is its need for an attribute-value representation. In general, graphs do not lend themselves to such a representation. Thus, future work includes implementing subgroup metrics directly into subgraph mining tools and evaluating them on real world data sets.

References

Herrara, F., Cristobal J.C., Gonzalez P., Jesus M.J. 2011. An overview of subgroup discovery: foundations and applications. *Knowl Inf Syst* 29:495-525

Gamberger D., Lavrač N. 2002. Descriptive Induction through Subgroup Discovery: A Case Study in a Medical Domain. ICML 2002: 163-170

Lavrač N., Flach P., Zupan B. 1999. Rule evaluation measures: a unifying view. *Proceedings of the 9th international workshop on inductive logic programming (ILP-99).* 1999:174-185

Cook D. J., Holder L.B. 2000. Graph-Based Data Mining. *IEEE Intelligent Systems* 15(2): 32-41

Gamberger D., Lavrač N. 2002. Expert-Guided Subgroup Discovery: Methodology and Application. *Journal of Artificial Intelligence Research* 17:501-527

Wrobel S. 2001. Inductive logic programming for knowledge discovery in databases. Springer, chap. Relational Data Mining, pp 74–101

Demšar, J., Curk, T., Erjavec, A. 2013. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research* 14:2349−2353

Krishna V., Ranga Suri N., Athithan G. 2011. A comparative survey of algorithms for frequent subgraph discovery. *Current Science (Bangalore)* 100(2):190-198

Orange – Data Mining Fruitful and Fun, Available at: *http://orange.biolab.si/*, (Accessed 2 October 2014).

Novak P., Lavrač N., Webb G. 2009. Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern, and subgroup mining, *Journal of Machine Learning Research*. 10: 377-403.