Use of frequent itemset mining for learning from graphs – what is gained and what is lost?

Thashmee Karunaratne and Henrik Boström

Department of Computer and Systems Sciences,
Stockholm University, Forum 100, SE-164 40 Kista, Sweden
{si-thk,henrik.bostrom}@dsv.su.se

Abstract. Graph mining methods have emerged to address the limitations of itemset mining algorithms when analyzing structured data. It may therefore appear counterproductive to employ the latter for mining graph data. Nevertheless, for graph classification tasks, where the focus is on predictive performance rather than comprehensibility, the use of itemset mining can be a sensible alternative to graph mining algorithms. In this paper, we examine the pros and cons of itemset mining on graph data using 18 medicinal chemistry datasets, and show that the itemset mining algorithms are not only efficient and reliable on graph classification and regression, but also competitive with the graph mining algorithms.

Keywords: Graphs, frequent itemset mining, classification, regression

1 Introduction

Methods that involve learning from graph databases broadly fall into three categories; graph similarity based methods [1], boosting methods [2] and mining sub-graphs from graph databases [3]. The latter, subgraph mining, methods differ from methods of the former categories due to their two-step approach of first discovering sub-graphs that possess high correlation with the target variable, and then use these sub-graphs as attributes to transform the graphs in the data base in to feature-vectors, which could be used together with standard machine learning algorithms. Methods of mining sub-graphs from graph databases include frequent sub-graphs and their representative subsets [4-7], interesting sub-graphs [8-10], significant sub-graphs [11], using methods such as sampling [12], pattern summarization [13], iterative feature selection [14] and so forth. These sub-graphs could be used within a spectrum of data mining tasks, such as classification, clustering, finding association rules, data indexing etc.

Sub-graph mining involves discovering sub-graphs and calculating their support. Discovery of sub-graphs could be apriori based or pattern-tree based. Apriori based [5] methods uses a breadth-first search for discovering candidate sub-graphs. In doing so, it starts with smaller sub-graphs and extends iteratively by increasing the size of the newly discovered sub-graphs by one node or edge. Usually, two sub-graphs of size k, i.e., graphs with k number of nodes, are joined together to form a subgraph of size $k+1$[3]. The essential drawback of this procedure is the overhead involved when joining two k sized graphs. The pattern-tree based approaches employ a depth-first search strategy [4,7]. In each iteration the pattern growth algorithm extends sub-graphs discovered during the previous iteration by one edge at every possible direction. Methods have been introduced to limit re-discovering the same graphs by techniques such as right-most expansion of the pattern tree [4]. Yet, it is a challenge to avoid the rediscovery of sub-graphs due to the repetition of node and edge labels in the graphs. The candidate graphs discovered by either of the approaches require counting their support, i.e., finding the number of occurrences of the sub-graphs in the graph database, which involves the subgraph isomorphism test, which is NP-Hard, or some canonical transformations to skip the isomorphism test that adds the transformation cost. Therefore, the graph mining problem could be complex depending on the size of the graphs.

Itemset mining algorithms discovers frequent or significant itemsets from transaction databases. Discovery of frequent itemsets is also a two-step procedure involving search of frequent patterns and calculation of support. The itemset mining algorithms use a lexicographical order of the items prior to mining and use horizontal [15] or vertical [16] layouts for organizing the transaction database to support efficient scanning, which leads to a simpler support count. The apriori principle [15] is usually employed to limit the search space during the itemset discovery. The lexicographical order of the items also helps avoiding re-discovery of the same itemsets. Therefore, despite of its exponential growth of complexity with respect to the number of items, the frequent itemset mining approaches are computationally simpler than their graph mining counterparts. Hence, the use of frequent itemset mining algorithms for graph mining, whenever applicable, could lead to significant computational savings. In [17], it was shown that frequent itemset mining could be efficiently used for mining a special type of graphs which are constrained by unique node and edge lables. A method that employs maximal frequent itemset mining on general graph data is discussed in [6]. Yet, rarely any study can be found that analyzes the effectiveness of itemset mining for graph classification. This study is motivated by such a need.

The rest of the paper is organized as follows. In the next section, the approach of using frequent itemset mining on graphs is described. In Section 3, an empirical investigation of using frequent itemset mining and graph mining on 18 medicinal chemistry datasets is presented. Finally, in section four, conclusions are given together with possible further extensions of this study.

2 Frequent itemset mining for graphs

A graph is a quintuple G= {V, E, λ ,Σ}, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges and $\lambda: V \cup E \rightarrow \Sigma$ is the labeling function. A graph $G_{s\,=}$ {V_s, E_s, λ_s, Σ_s} is said to be *subgraph isomorphic* to *G*, which is denoted by $G_s \subseteq G$, if there exists a 1–1 mapping $f: V_s \rightarrow V$ such that, $\forall v \in V_s$, $\lambda_s(v) = \lambda(f(v))$ and, $\forall(v_i,v_j) \in E_s$, $(f(v_i), f(v_j)) \in E$ and, $\forall(v_i, v_j) \in E'$, $\lambda'(v_i, v_j) = \lambda(f(v_i), f(v_j))$. Further, we say G_s *occurs* in *G* if $G_s \subseteq G$. Let the database D contain a collection of graphs *G*, then, the *support* of a subgraph G_s in D is the number of *occurrences* of G_s in D.

A graph may be transformed into an *edge list* L of G = {V, E, λ, Σ}, in order to allow for applying itemset mining algorithms. An edge list is defined as $L = (v_i, v_{j\,,}\, e_k \mid \forall v_i,v_j \in V$ and $e_k \in (v_i, v_j))$. Let $l \in L$, then $\forall l \in L$ are distinct if and only if $\lambda: V \cup E \rightarrow \Sigma$ is injective. Within this framework, a graph mining problem can be viewed as a frequent itemset mining problem such that; let L be the set of items and $X \subseteq L$ be an itemset. Let the transaction database D be a multiset of subsets of L. For itemset *X*, a transaction including X is an *occurrence* of X and the *support(X)* is the percentage of any itemsets $Y \subseteq X$ over the transaction database. The frequent item set mining determine all the itemsets X such that *support(X)* ≥ *minimum support* given a *minimum support*. X is a *maximal frequent itemset* when X is included in no other frequent itemset.

In this study, we employ the MFI (**M**aximal **F**requent **I**temset) algorithm [6], which uses maximal frequent itemset mining for discovering frequent sub-graphs (which are not necessarily connected) from graph databases. The MFI algorithm requires transformation of graph data into edge lists. The MAFIA algorithm [16] is used on the edge lists to discover the maximal frequent itemsets. Another application of itemset mining that we consider in this study is based on itemset mining using constraint programming. Similar to MFI, the edge lists of the graphs are used in the CPIM algorithm [18] to mine significant patterns.

We also consider three graph mining methods, namely, **GraphSig** (Mining Significant graphs) [11], **MoFa** (frequent molecular fragments miner) [7], **SUBDUE** (interesting sub-graphs discovery) [10].

3 Empirical Evaluation

In this experiment, we use eighteen datasets from the medicinal chemistry domain, which are publicly available [19]. The discovered sub-graphs from the graph and itemset mining methods are used as features for building the predictive models. The same datasets are used for investigating both classification and regression performance.

Classification models are built using random forests (RF), support vector machines with the RBF kernel with complexity 2 (SVM-R), and the polynomial kernel with complexity 2 (SVM-P), and the k-nearest neighbor (KNN) as implemented in the WEKA data mining toolkit [20]. We considered accuracy as the performance criterion, which is estimated using 10-fold cross-validation. Since the intention of the study is to draw conclusions of the relative performance of the descriptor sets without reference to a specific learning algorithm, we randomly choose one of the three learning algorithms for each dataset, and draw one conclusion for all descriptor sets independently of the algorithms. The alternative would be to apply all learning algorithms on all descriptor sets, and either draw one conclusion for each learning algorithm or look for the best combination of descriptor set and learning algorithm, something which would typically require a very large number of datasets in order to allow for any statistically significant differences to be detected.

For regression, we have used the support vector machine with the nonlinear polynomial kernel with complexity 2 and the RBF kernel with complexity 2, as implemented in WEKA [20]. The root mean squared error (RMSE) where chosen as performance criterion for the regression tasks, again using 10-fold cross validation. The class labels are used for feature construction in MoFa, GraphSig, SUBDUE, and CP. Parameter optimizations of the algorithms, when required, were done by cross-validation on the training sets, where the optimized parameters were used in the test set. The same training and test folds were used for all methods. Again, we choose one of the learning algorithms randomly for each dataset.

The experiments were carried out using a HP EliteBook, with two Core2 Duo Intel processors 2.40GHz each, and 2.9GB main memory, running under Ubuntu 10.4. An upper limit of 24 hours was set for each method on each dataset.

3.1 Results

Table 1 summarizes the datasets and shows the average time taken by each method on sub-graph discovery. In Fig. 1, the classification accuracies of the models are shown. Due to the space constraints we omit including the regression models.

Table 1. Average time taken (in seconds) by each method for sub-graph discovery

Dataset Name	#graphs in the dataset	Average graph size	#node labels	graphSig	MFI	MoFa	CP	SUBDUE
ace	114	42	7	292.81	0.41	11.71	1.40	2.54
ache	111	56	7	-	0.87	-	7.16	43.54
AI	69	24	8	1041.73	0.41	45.57	4.33	383.71
AMPH1	130	87	5	114.63	1.01	-	11.35	27.10
ATA	94	22	5	8.50	0.50	6.88	2.50	6.33
bzr	163	36	8	-	1.00	285.15	4.36	6.87
caco	100	45	8	2.37	0.70	147.57	0.75	1.37
COMT	92	20	7	2.73	0.67	2.97	2. 72	3.33
cox2	322	41	8	3000.03	2.00	2136.92	2.31	14.9
dhfr	397	41	8	1135.15	2.38	22.63	4.48	8.37
EDC	119	19	7	0.35	0.72	3.98	0.65	4.50
gpb	66	32	8	1878.64	0.53	8.13	0.82	1.42
HIVPR	113	45	9	1414.77	0.82	2558.54	1.44	3.41
HIVRT	101	25	9	85.94	0.67	3.24	1.16	2.10
HPTP	132	38	9	2573.07	0.85	192.28	0.69	1031.50
nct	131	20	8	255.98	0.80	21.33	1.81	8.50
therm	76	52	6	196.38	0.59	28.12	0.62	11.21
thr	88	68	5	1944.58	0.71	-	0.59	51.51

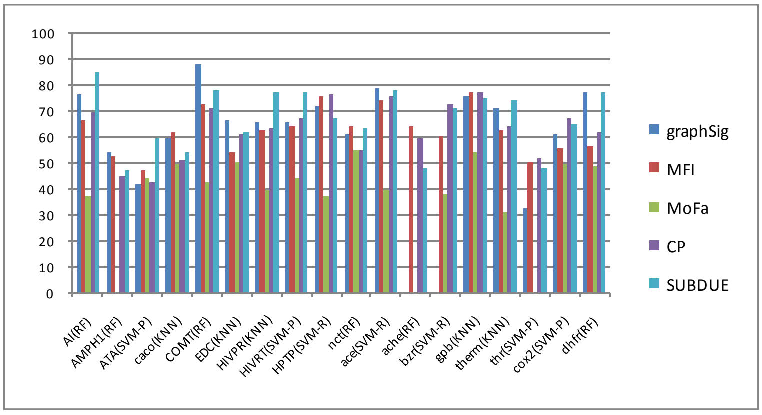

Fig. 1 Classification accuracies

To evaluate the results statistically, a null hypothesis is formed stating that there is no significant difference between the accuracies obtained by using different feature construction methods. The significance of the differences of the regression errors and the classification accuracies is tested by comparing the ranks (relative performance of the methods) using the Friedman test [21]. The Friedman test rejected the null hypothesis for both regression and classification experiments. Therefore, further tests were conducted to identify pairs of methods for which the difference in performance is significant. The average ranks are used for pair-wise tests for significance, based on the Nemenyi test [21]. In applying this criterion, a method that fails to produce a feature set is assigned the highest rank, which corresponds to the worst performance.

Table 2 gives the differences of ranks for all the pair-wise tests of the regression models and Table 3 represents the same for classification models. The pairs corresponding to dark cells in the tables show the methods that are significantly different in their performance, i.e., the differences of ranks are larger than the Critical Difference [21] 1.44 (a positive value corresponds to that the method in the row label outperforms the method in the column label and vice versa for negative values). According to Table 2 and 3, all significant differences involve MoFa.

Table 2. Differences of average ranks of performance of regression models

	graphSig	MFI	MoFa	CP	SUBDUE
graphSig	-				
MFI	-1.14	-			
MoFa	-1.64	-0.50	-		
CP	-0.08	1.06	1.56	-	
SUBDUE	0.25	1.39	1.89	0.33	-

Table 3. Differences of average ranks of performance of classification models

	graphSig	MFI	MoFa	CP	SUBDUE
graphSig	-				
MFI	-0.22	-			
MoFa	-2.22	-2.00	-		
CP	-0.06	0.17	2.17	-	
SUBDUE	0.39	0.61	2.61	0.44	-

The experiment shows that the relatively high computational cost of graph mining methods compared to the itemset mining approaches is not motivated by a corresponding gain in predictive performance. On the contrary, it can be seen that the itemset mining approaches significantly outperform one of the graph mining approaches. Hence, in cases where computational cost is important, the experiment shows that itemset mining approaches can be good alternatives to graph mining approaches for prediction tasks. What is lost when employing itemset mining compared to graph mining is the topological structure of graphs, i.e., the identified sub-structures cannot be easily interpreted. However, when predictive performance is all that matters, this loss is negligible.

4 Concluding remarks

Itemset mining algorithms are not as expressive as graph mining methods since they have less power to encode the topological structure of graphs. On the other hand, itemset mining algorithms are computationally less complex. In this study, we have investigated the effectiveness of using itemset mining compared to graph mining for graph prediction tasks.

An experiment with building predictive models for classification and regression tasks using 18 medicinal chemistry datasets was presented. It was concluded that models built using features discovered by itemset mining algorithms were, despite less expressive power and requiring much lower computational cost, not only competitive with the standard graph mining methods, but also that they even may outperform some graph mining methods.

The presented experiment focused on graph data from the domain of medicinal chemistry. An immediate question is whether or not the conclusions from this study carry over to other domains as well. The study could, for example, be extended towards specific domains of graphs with unique node labels and compare the performances with graph mining methods, since unique node graphs preserve the topological structure of the graphs under the canonical transformation of graphs using edge lists.

References

1. Hinselmann G Ã, Fechner N, Jahn A, Eckert M, Zell A, Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments, Neurocomputing 74, 2010, pp 219–229
2. Kudo T, Maeda E and Matsumoto Y, An application of boosting to graph classification, *NIPS'04*, 2004, pp 729 – 736
3. Jiawei H, Hong C , Xin D and Yan X, Frequent pattern mining: current status and future directions, Journal of Data Mining and Knowledge Discovery, 1384-5810, 15(1), 2007
4. Yan X. and Han J., gSpan: Graph-Based Substructure Pattern Mining, *ICDM'02*, 2002
5. Kuramochi M and Karypis G, Frequent subgraph discovery. *ICDM'01*, pp 313–320, 2001
6. T. Karunaratne and H. Boström, "Graph Propositionalization for random forests", *ICMLA'09*, 2009, pp 196-201
7. Borgelt C and Berthold M, Mining Molecular Fragments: Finding Relevant Substructures of Molecules, *ICDM 2002*, 2002, pp 51-58
8. H. Cheng, X. Yan, J.Han and C. Hsu, Discriminative frequent pattern analysis for effective classification, *ICDE'07*, 2007, pp 716-725
9. N. Jin, C. Young and W. Wang, GAIA: graph classification using evolutionary computation, *SIGMOD '10,* 2010, 879-890
10. N. S. Ketkar, L. B. Holder and D. J. Cook, Subdue: Compression-Based Frequent Pattern Discovery in Graph Data, *OSDM '05*, 2005, pp 71-76
11. Ranu S, and Singh A K. GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases, *ICDE'09*, 2009, pp.844-855
12. X. Yan, X.J. Zhou and J. Han, Mining closed relational graphs with connectivity constraints. *KDD'05,* 2005, pp 324–333
13. M Hasan and M. J. Zaki. MUSK: Uniform Sampling of k Maximal Patterns, *SDM'09*, *2009*, pp.650-661
14. H. Saigo, N. Krämer and K. Tsuda, Partial least squares regression for graph mining, *KDD '08*, 2008, pp 578-586
15. Agrawal R., Srikant R. and Swami A., Mining Association Rules between Sets of Items in Large Databases, *SIGMOD'93*, 22(2) , 1993, pp. 207-216
16. D. Burdick, M. Calimlim and J Gehrke, MAFIA: a maximal frequent itemset algorithm for transactional databases, *ICDE'01*, 2001, pp 443–452
17. L.Thomas, Maximal frequent subgraph mining, Phd Thesis, International Institute of Information Technology,Hyderabad, India, 2010
18. S. Nijssen, T. Guns and L. De Raedt, Correlated itemset mining in ROC space: a constraint programming approach, *KDD '09*, 2009, pp 647-656
19. Data Repository of Bren School of Information and Computer Science, University of California, Irvine, ftp://ftp.ics.uci.edu/pub/baldig/learning/, 2005, Last visited: 18/04/2011
20. I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, 2nd Edition*, Morgan Kaufmann, San Francisco, USA, 2005
21. Garcia S and Herrera F, An Extension on "Statistical comparisons of classifiers over multiple data sets" for all Pairwise Comparisons, *Journal of Machine Learning Research*, 9 (2008), 2677-2694