Graph Propositionalization for Random Forests

Thashmee Karunaratne

Dept. of Computer and Systems Sciences,
Stockholm University
Forum 100, SE-164 40 Kista, Sweden
si-thk@dsv.su.se

Henrik Boström

Dept. of Computer and Systems Sciences,
Stockholm University
Forum 100, SE-164 40 Kista, Sweden
henrik.bostrom@dsv.su.se

***Abstract* - Graph propositionalization methods transform structured and relational data into fixed-length feature vectors that can be used by standard machine learning methods. However, the choice of propositionalization method may have a significant impact on the performance of the resulting classifier. Six different propositionalization methods are evaluated when used in conjunction with random forests. The empirical evaluation shows that the choice of propositionalization method has a significant impact on the resulting accuracy for structured data sets. The results furthermore show that the maximum frequent itemset approach and a combination of this approach and maximal common substructures turn out to be the most successful propositionalization methods for structured data, each significantly outperforming the four other considered methods.**

Key words – graph propositionalization, random forests, structured data

I. INTRODUCTION

Preprocessing of data is an essential requirement for many learning tasks, since most of the standard machine learning algorithms require data to be represented in an attribute-value format where the instances are represented using feature vectors of fixed length [1]. However, in many cases, relevant information cannot easily be represented in a single table, since relations between the attributes are frequently occurring [2]. Therefore, in order to apply the learning algorithms in such domains, either the relational aspects have to be omitted or a suitable transformation of the relations into the attribute-value format is required. To avoid loss of information, the latter approach should be adopted if possible, hence requiring an appropriate preprocessing method for including the structural information into the attribute-value format.

Propositionalization methods have achieved popularity within the machine learning community for their ability to allow for learning using adapted propositional algorithms [3,4,5]. A typical propositionalization model employs a feature vector representation where the elements of the feature vector include the relations [5]. A relational learner based on propositionalization uses a transformation module to compute a set of propositional features and then uses a (standard) machine learning algorithm for model development [6]. Graph propositionalization methods are a subclass of these methods that consider examples represented by graphs and by some suitable approach convert the graphs into fixed-length feature vectors.

Several approaches for building classification models using propositionalization can be found in the literature [4,5,6,7,8,9,10]. The propositionalization methods are usually embedded in a discovery, prediction or classification method, and therefore are difficult to view as a data preprocessing method. Hence, it has so far not been carefully investigated which propositionalization methods are best suited for different standard machine learning algorithms.

Among the standard machine learning algorithms, random forests have been shown to be robust and computationally efficient with excellent predictive performance [11,12]. Therefore we have chosen random forests as a state-of-the-art learning algorithm when investigating the effectiveness of different graph propositionalization methods.

The paper is organized as follows. The next section presents different graph propositionalization approaches that are considered in this study. Section three presents an empirical evaluation of these methods when used in conjunction with random forests. Section four reports the outcome of the experiments. Main conclusions from the study and directions for future work are finally given in section five.

II. GRAPH PROPOSITIONALIZATION METHODS

The graph propositionalization methods used in this study assume that the graphs are given in a canonical form

(called *fingerprint*) and several methods to extract fixed-length feature vectors from these are investigated.

Graphs are assumed to be represented by nodes (e.g., an atom in a molecule) and edges (e.g., a bond connecting two atoms). Furthermore, it is assumed that all nodes have been given labels, allowing similar nodes in different graphs to be handled in a similar way (e.g., an atom could be given the label 'carbon'). Each example is represented by the set of all triples *(Li,Lj,Ek),* such that there is an edge labeled E_k in the graph of the example between nodes N_i and N_j that are labeled L_i and L_j respectively. Following [8], such a set is referred to as a *fingerprint*.

A. Feature extraction methods

In this study, we consider five different feature extraction methods for propositionalization from a given fingerprint representation. The five methods, which are described below, are based on the vector space model, maximum frequent itemsets, maximal common substructures, information gain, and a combination of maximum frequent itemsets and maximal common substructures, respectively.

1) The vector space (vs) model

The vector space model is quite popular within the text classification community for its simplicity. This model is used in information retrieval, where each document is represented by a vector of terms in the document collection. Therefore, if a document collection consists of the set of terms $T = \{t_1, \ldots, t_n\}$, an arbitrary vector for a particular document could be represented as $\underline{d_i} = \{ w_1, \ldots, w_n\}$ where each w_i corresponds to the weight of the respective term t_i within the d_i^{th} document. There are several approaches to obtain the weights, yet term frequency is the most common and simplest among them.

The fingerprint of a certain graph could be interpreted as a document that consists of a set of terms. Therefore considering each triple (L_i,L_j,E_k) as a term, we could represent a fingerprint by a term frequency vector. Here the weights of the elements of the feature vector is the frequency of (L_i,L_j,E_k) in each fingerprint

2) Maximum frequent itemsets (mfi)

An Apriori based method for classification was first introduced in [13], and has gained considerable popularity since then. The Apriori method is based on analyzing common item sets in a set of transactions, which is technically referred to as frequent itemset mining. The search space of this problem, which is all frequent item sets, is typically huge. Several approaches to efficiently explore the search space have been proposed that only consider a sufficiently large subset of all frequent itemsets. Among these, methods for finding the set of all closed or maximal itemsets are the most popular [14].

Given below is the definition of frequent itemset mining and maximal frequent itemset mining [14,15]:

Definition: *maximal frequent itemset*: Suppose we are given a set of items I, which is a set of literals. An itemset $I \subseteq \mathcal{I}$ is some subset of items. For itemset $\mathcal{I}$, a transaction including I is called an occurrence of I. The *denotation* of I, denoted by $T(I)$ is the set of the occurrences of I. $|T(I)|$ is called the frequency of I; and denoted by $frq(I)$: For a given constant θ, called a *minimum support*, itemset I is frequent if $frq(I) \geq \theta$. If a frequent itemset I is included in no other frequent itemset, I is called *maximal*.

When adapting this method for feature selection, the item set $I \subseteq \mathcal{I}$, where, $\mathcal{I}$ is the collection of all triples (L_i,L_j,E_k) in the input graphs as described above, are considered in a maximal frequent item set mining algorithm as defined in [15]. The discovered maximum frequent itemsets are treated as features to the classifier.

3) Maximal common substructures (mcs)

The finger prints generated by the graph transformation method are considered in the maximal common substructure search algorithm [8]. The procedure for the maximal common substructure search is as follows. For all pairs of examples, the intersection of their fingerprints, which is referred to as *the maximal common substructure,* is formed, and ranked according to their frequency in the entire set of examples (i.e., the number of fingerprints for which the maximal common substructure is a subset). Upper and lower thresholds are applied to select useful substructures for classification [8].

4) Information gain (ig)

The maximal common substructure search algorithm described above does not make use of class labels when selecting the features, but only ensures that selected features fulfill the user-defined upper and lower bounds. Instead of relying on these bounds, another method is proposed that calculates the information gain of each of the features generated by the maximal common substructure search algorithm and chooses the n most informative features, where n is a parameter of the method.

5) Maximum frequent itemsets + maximal common substructures (mfi+mcs)

This method first generates the original maximal common substructures by considering the fingerprints and the resulting pair-wise maximal common substructures (mcs) are treated as an item set, so that the maximal frequent itemset mining could produce a maximal common itemset (mfi). This itemset will be the selected feature set. In doing so we have taken all the substructures generated by the maximal common substructure search method, without applying any threshold to select a subset of it.

III. EMPIRICAL EVALUATION

We have set up an experiment to test the null hypothesis that different propositionalization methods perform equally well in conjunction with random forests. In addition to the methods described in the previous section, we have also used a base-line representation to preprocess the inputs to the random forest algorithm. A detailed description of the six different preprocessing methods is given below.

The Random Forest classifier, as implemented in the WEKA data mining toolkit [16], is used for the entire set of experiments carried out in this study. Each generated random forest consists of 50 trees and the accuracy of each method is estimated using 10-fold cross-validation.

attribute value representation (av) - The standard attribute-value format by which the data is represented in a matrix where the columns correspond to the attributes and the rows represents the values of the attributes. Attributes in this representation are the nodes of the graphs and the corresponding values represent the frequencies of the nodes in each graph.

vector space representation (vs) - All fingerprints are transformed into a term-frequency matrix, where each row of the matrix represents the documents (fingerprints) and columns represent the term frequencies.

maximum frequent itemsets (mfi) - The maximum frequent itemsets (mfi) are selected from the set of fingerprints. We have used the maximal frequent itemset mining algorithm as implemented in MAFIA [17], together with 6 levels of support, namely 0.5, 0.4, 0.2, 0.1, 0.025 and 0.01. The level of support resulting in the highest accuracy, as estimated by 10-fold cross-validation on the training set, is used for generating the model that is evaluated on the test set.

maximal common substructures (mcs) - The maximal common substructure search algorithm as implemented in DIFFER [8] is used for preprocessing the input to the Random Forest classifier.

information gain (ig) - Information gain is calculated on the full set of substructures discovered by the maximal common substructure search algorithm [8]. The best 10, 20, 50 and 100 substructures are used in model building. Again 10-fold cross-validation on the training set is used to determine what number of substructures to use when generating the model that is evaluated on the test set. The information gain criterion, as implemented in WEKA [16] is used for the calculation.

maximal frequent itemset mining + maximal common substructures (mfi+mcs) - DIFFER's mcs algorithm [8] is merged with MAFIA's mfi algorithm [17]. We have used the same set of possible levels of support as for mfi. The support level resulting in the highest accuracy, as estimated by 10-fold cross-validation on the training data, is used to generate the model that is evaluated on test data.

A. Null hypothesis

The null hypothesis is that there is no difference between the propositionalization methods when used in conjunction with random forests.

B. Data sets

We have used 30 publicly available data sets to test the null hypothesis. Of these datasets, 15 are structured, i.e., the data contains relations among attributes and the remaining 15 datasets are non-structured, i.e., there are no specified relations among the attributes. While the structured data are transformed into graphs according to the definitions given in [8], the non-structured data are converted into virtual graphs in such a way that the attribute values are represented by node labels and the attribute names are represented by edge labels. For example "a day with a high temperature and low humidity" would be represented by a graph as in the figure 3 given below.

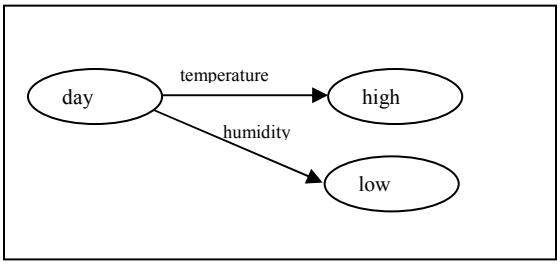

Figure 1: Virtual graph for non-structured data.

IV. EXPERIMENTAL RESULTS

The results from employing 10-fold cross-validation on the 30 data sets using the standard random forest classifier in conjunction with the six different representations are shown in Table I and II below.

A. Hypothesis tests

In testing the null hypothesis, we have considered the structured and non-structured datasets separately.

1) Structured data sets

The null hypothesis, i.e., there is no difference in classification accuracy between the six methods on structured data, is tested using the Friedman test [18].

The average ranks for the six methods strongly suggest rejecting the null hypothesis, i.e., there is no difference in predictive performance among the methods, as the F-statistic 19.77 is greater than the tabulated value for $F_{(5,70)}$, i.e., 1.90. Also, the results shown in Table III indicate that two distinct groups of classifiers can be identified.

TABLE III. AVERAGE RANKS FOR STRUCTURED DATASETS

Method	av	mfi	mfi+ mcs	ig	vs	mcs
Average rank	5.00	2.03	2.17	3.5	4.13	3.87

According to the Table III, the methods *av* and *vs* stand out as rather poor methods, while *mfi* and *mfi+mcs* stand out as performing relatively well.

After concluding that there is indeed a difference in the observed performance among the methods, we examine the differences in performance when pair-wise comparing the six propositionalization methods, using the Nemenyi test [18]. The differences in average ranks for each pair of methods are shown in Table IV.

TABLE IV. DIFFERENCE OF AVERAGE RANKS

Method	av	mfi	mfi+mcs	ig	vs	mcs
av	-					
mfi	2.97	-				
mfi+mcs	2.83	0.14	-			
IG	1.5	1.47	1.33	-		
vs	0.87	2.1	1.96	0.63	-	
mcs	1.13	1.84	1.7	0.37	-0.26	-

According to the post-hoc Nemenyi test [18], the performance of two classifiers is significantly different, i.e., at the 0.05 level, if the corresponding average ranks differ by at least the critical difference, which in this case is 1.33. Therefore the null hypothesis that a pair of methods result in equally accurate models can be rejected for the following pairs: (av, mfi), (av, mfi+mcs), (av, ig), (mfi, ig), (mfi, vs), (mfi, mcs), (mfi+mcs, vs) and (mfi+mcs, mcs).

2) Non-structured datasets

We have tested the null hypothesis also for non-structured data (Table II). The average ranks obtained for the non-structured dataset is given in Table V below.

TABLE V. AVERAGE RANKS FOR NON-STRUCTURED DATASET

Method	av	mfi	mfi+ mcs	ig	vs	mcs
Average rank	3.87	3.87	2.97	3.70	2.6	3.97

According to the Friedman test [18], the null hypothesis cannot be rejected in this case, since the F statistic is $0.035 < F_{(5,70)}(2.3683)$.

Hence, the results indicate that there is nothing to gain from applying graph propositionalization when the data is essentially unstructured.

V. CONCLUDING REMARKS

Graph propositionalization methods can be used with random forests as they allow for transforming graph data into the form of fixed-length feature vectors. In this paper, six propositionalization methods are evaluated in conjunction with random forests. An empirical evaluation on 30 data sets, of which 15 are structured, was presented, showing that the choice of propositionalization method has a significant impact on the resulting accuracy for structured data sets.

The maximum frequent itemset (mfi) approach and maximum frequent itemsets combined with maximal common substructures (mfi+mcs) turn out to be the most successful propositionalization methods for structured data, each significantly outperforming all other four methods.

One direction for future work is to develop and evaluate additional graph propositionalization methods for random forests. Another direction would be to investigate whether similar findings regarding the relative performance of the investigated propositionalization methods also hold for other state-of-the-art machine learning methods, such as support-vector machines and boosted decision trees.

REFERENCES

[1]. Eibe Frank, Mark A.Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, and I. H. Witten, "Weka: A machine learning workbench for data mining", *Data Mining and Knowledge Discovery Handbook,* 978-0-387-24435-8© Springer, Berlin, 2005

[2]. Nicolas Lachiche, "Good and Bad Practices in Propositionalisation". *AI*IA 2005*: 3673. Springer 2005, ISBN 3-540-29041-9 50-61

[3]. I.Bournaud, M. Courtine, and J. Zucker, "Propositionalization for Clustering Symbolic Relational Descriptions", S. Matwin and C. Sammut (Eds.): *ILP 2002*, LNAI 2583, pp. 1–16, 2003, Springer

[4]. ,M.-A. Krogel, and S. Wrobel, "Transformation-based learning using multirelational aggregation", In C. Rouveirol and M. Sebag, editors, *Inductive Logic Programming*, 11th International Conference, volume 2157 of Lecture Notes in Computer Science, 2001, pages 142–155. Springer

[5]. S.Kramer, N. Lavrač and P Flach, "Propositionalization approaches to relational data mining", In S. Dězeroski, Ed. *Relational Data Mining*, Springer-Verlag New York, 2000, 262-286

[6]. Mark-A. Krogel, Simon Rawles, Filip Železný, Peter A. Flach, Nada Lavrač and Stefan Wrobel, "Comparative Evaluation of Approaches to Propositionalization", *Inductive Logic Programming*, Lecture Notes in Computer Science Volume 2835/2003, 2003

[7]. F. Železný and N. Lavrač, , "Propositionalization-based relational subgroup discovery with RSD", *Machine Learning,* Volume 62, Numbers 1-2 / February, 2006, 33-63, Springer

[8]. T. Karunaratne and H. Boström, "Learning to Classify Structured Data by Graph Propositionalization", Proceedings of the Second *IASTED International Conference on Computational Intelligence*, 2006, 393-398

[9]. S. Kramer, "Relational learning Vs. propositionalization", *AI communications*, ISSN 0921-7126, 215-281, IOS press

[10]. L. Breiman, J. H. Friedman, R. A. Olshen and C. J .Stone, "Classification and Regression Trees", *Wadsworth*, Belmont, 1984

[11]. R. Caruana and A. Niculescu-Mizil, , "An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics", *Proceedings of 23rd Intl. Conf. Machine learning -ICML'06*, 2005, Springer, 161-168

[12]. L. Breiman, "Random forests", *Machine Learning*, 2001, 45:5-32.

[13]. R. Agrawal and R. Srikant. "Fast algorithms for mining association rules". In 20th VLDB Conference, Sept. 1994, 487-499

[14]. B. Goethals and M. J. Zaki. "FIMI'03: Workshop on frequent itemset mining implementations. In B. Goethals and M. J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI2003), volume 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, 19 November 2003

[15]. T. Uno, M. Kiyomi, and H. Arimura. LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets. In B. Goethals, M. J. Zaki, and R. Bayardo, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2004), volume 126 of CEUR Workshop Proceedings, Brighton, UK, 1 November 2004

[16]. Ian H. Witten and Eibe Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, USA, 2005

[17]. Doug Burdick, Manuel Calimlim and Johannes Gehrke., "MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Database" *In Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, Germany, April 2001

[18]. J. Dem˘sar. "Statistical comparisons of classifiers over multiple data sets", *Journal of Machine Learning Research*, 7:1–30, 2006

[19]. Data Repository of Bren School of Information and Computer Science, University of California, Irvine, ftp://ftp.ics.uci.edu/pub/baldig/learning/QSAR_Sutherland/, 2005

[20]. Muggleton, S., Data sets for Progol and Golem, http://www.doc.ic.ac.uk/~shm/Software/Datasets/, 2005

[21]. C. Blake and C. Merz. UCI repository of machine learning databases, 1998

[22]. SUBDUE data repository, http://ailab.wsu.edu/subdue/, as at 26th June 2009.

TABLE I. RESULTS FOR STRUCTURED DATASET

Data set – Structured	av	mfi	mcs + mfi	IG	vs	mcs
ace (2cl) [19]	54.87	66.14	65.38	64.14	57.50	63.72
ache (2cl) [19]	46.36	66.36	65.46	61.62	47.27	60.90
chang (3cl) [19]	58.82	68.33	68.33	58.80	58.82	60.00
bzr (3cl) [19]	59.50	56.91	56.88	58.75	58.28	60.12
mutagenesis (2cl) [20]	74.25	85.60	86.17	83.89	86.89	80.85
carcinogenesis (2cl) [20]	64.10	65.78	64.11	62.30	62.96	60.90
suramin (2cl) [20]	70.00	81.82	81.82	65.00	63.64	65.00
huuskonen (3cl) [19]	50.75	59.53	59.53	58.80	74.06	63.60
ProStu (2cl) [21]	79.30	86.67	86.96	80.50	80.44	80.43
cox2 (3cl) [21]	55.90	62.48	61.56	58.13	56.21	58.38
dhfr (2cl) [21]	51.64	58.17	59.46	59.69	56.63	59.69
CompProf (2cl) [21]	53.80	60.59	59.76	58.00	48.15	51.85
crist (4cl) [19]	37.50	46.67	45.00	36.67	46.88	36.67
gpb (2cl) [19]	50.00	63.33	58.80	58.28	57.58	54.55
thr (2cl) [19]	61.36	63.75	64.86	61.47	57.96	64.77

TABLE II. RESULTS FOR NON-STRUCTURED DATASET

Data set – non Structured	av	mfi	mcs + mfi	IG	vs	mcs
spect (2cl) [21]	69.66	81.25	81.27	76.01	80.90	80.15
lens (3cl) [21]	70.83	80.50	80.50	80.33	75.00	70.83
monk (2cl) [21]	91.80	90.98	91.80	93.33	93.92	92.62
shuttle (2cl) [21]	66.67	55.00	70.00	66.67	66.67	66.67
hayes (3cl) [21]	75.76	80.22	81.82	81.06	87.12	81.06
balance (3cl) [21]	77.92	83.44	83.52	81.75	81.76	83.84
balloons (2cl) [21]	100.00	100.00	100.00	97.48	100.00	100.00
cars (4cl) [21]	94.33	96.41	96.88	95.43	96.70	95.36
golf (2cl) [22]	57.14	55.00	55.00	55.37	50.00	50.00
voting (2cl) [22]	95.40	94.71	94.73	95.12	95.63	93.56
diabetes (2cl) [22]	68.09	69.53	69.79	72.79	68.62	66.67
credit (2cl) [22]	74.10	72.00	69.10	70.60	73.50	69.80
nursery (5cl) [21]	92.94	87.09	88.71	92.74	92.74	92.74
Post-operative (3cl) [21]	62.22	63.33	65.56	60.67	65.56	64.44
dermatology (6cl) [21]	95.90	95.92	96.23	90.23	97.27	92.62