LEARNING TO CLASSIFY STRUCTURED DATA BY GRAPH PROPOSITIONALIZATION

Thashmee Karunaratne

Department of Computer & Systems Sciences (DSV)
Stockholm University and Royal Institute of Technology
Forum 100, SE 164 40, Kista
Sweden
si-thk@dsv.su.se

Henrik Boström

Department of Computer & Systems Sciences (DSV)
Stockholm University and Royal Institute of Technology
Forum 100, SE 164 40, Kista
Sweden
henke@dsv.su.se

ABSTRACT
Existing methods for learning from structured data are limited with respect to handling large or isolated substructures and also impose constraints on search depth and induced structure length. An approach to learning from structured data using a graph based propositionalization method, called finger printing, is introduced that addresses the limitations of current methods. The method is implemented in a system called DIFFER, which is demonstrated to compare favorable to existing state-of-art methods on some benchmark data sets. It is shown that further improvements can be obtained by combining the features generated by finger printing with features generated by previous methods.

KEY WORDS
Machine Learning, Graph, Classification, Structured data

1. Introduction

In many domains, in which a model is to be generated by machine learning, examples are more naturally represented by structured terms than fixed-length feature vectors. For example, in chemo-informatics, molecules are naturally represented as two or three dimensional structures of atoms. Another example is when having data on XML format, which could be directly mapped on tree structures.

Several approaches to learning to classify structured data have been introduced in the field of machine learning. The structure classification problem has been addressed as a rule learning problem [1,2,3,4,5], as a graph mining problem [6,7,8,9,10,11,12,13,14,15] and as a propositionalization problem [16,17,18]. These methods address two main varieties of problems. The first category concerns discovery of features that best discriminate between different classes. Krogel et al [16], for example, introduce an approach to selecting the "most interesting" features of structured data that discriminate between the classes. A key requirement of the feature discovery methods is that the discovered features should be comprehensible.

In contrast to the discovery methods, classification methods generate global models for classifying all examples, but the models need not necessarily be comprehensible. Most classification methods assume that all examples can be represented by fixed-length feature vectors, and finding features that suitably contain the relevant information in this format can be considered a major knowledge engineering bottleneck for these methods. This is true in particular when the examples are most naturally represented as structured terms (e.g., trees, lists, etc.). Existing methods for structure classification are limited with respect to finding large or isolated substructures or by requiring constraints on search depth and size of substructures considered, as further described in section 2, and hence more robust methods for learning from structured data are needed. The method presented in this paper, which extracts features from structures by a method called finger printing, is motivated exactly by this need.

The rest of the paper is organized as follows. In section 2, the state-of-art structure classification methods are discussed together with their limitations. The novel finger printing method, that addresses these limitations, is introduced in section 3. In section 4, an empirical evaluation is presented, comparing the novel method to state-of-the-art methods on some benchmark datasets. Finally, in section 5, we give concluding remarks and outline possible further extensions to this study.

2. Current Approaches to Learning from Structured Data

Current state-of-art methods for feature discovery and classification use several forms of structure transformation. Inductive logic programming [5] has drawn immense popularity since its inception, mainly due to that background knowledge and data as well as the result of the methods are represented in the same format: logic programs. Propositionalization methods is one class of ILP methods that transform the relational rule learning problem into a standard attribute-value learning problem by identifying suitable features [16]. However, these, as well as the standard ILP methods, are often faced with a huge search space, either for which constraints have to be imposed, or for which the domain has to be restricted in terms of the number of examples considered [19]. The limits on search depth and clause length typically result in that the substructures discovered by ILP methods are quite small and usually are limited to 5-6 structural relations [20].

Graph mining methods, including kernel methods, are efficient enough to discover considerably larger substructures compared to the ILP methods [9,20]. Although the current algorithms already perform quite well, they still have some limitations. The graph mining approaches suffer from the decidability problem of isomorphism between sub-graphs, which is NP-complete [20]. Graph kernels [12,14] have been demonstrated to result in accurate classifiers, but defining an appropriate kernel function for a particular problem remains a challenge. " It is known that computing complete graph kernels is at least as hard as deciding whether two graphs are isomorphic" [13]. Kernal methods consider parts of graphs such as walks, cyclic paths etc. in defining kernel functions and therefore it is a challenge for kernel methods to consider "the entire structure of the graph into account. While those kernels can be defined, computing them is hard" [13]. Also the discovered graphs by kernels or frequent graph mining methods are required to be connected by necessity. This prevents inclusion of isolated or far away frequent nodes or sub graphs. Thus two fragments within a graph that are not connected are not being considered in conjunction by the current methods, even if the contribution of these fragments when taken together would be a highly potential feature. Another limitation of current graph mining methods is that they only consider exact matches of the sub-graphs and hence do not allow mining *similar sub-graphs* [21], i.e., sub-graphs that are not exactly equal to each other (also referred to as inexact sub-graphs), but differ only by a few nodes. For example, in a chemo-informatics application, different molecules may have carbon chains of different lengths, but to which the same topology of atoms may be connected, i.e., the corresponding substructures differ only by its length of the carbon chain. These substructures are not exactly equal to each other since they differ by the length of the carbon chain, but rather *similar* since the topology other than the length of the carbon chain of the substructures is the same. Current methods consider these substructures as completely different, since the substructures do not exactly match with each other. A further discussion about similar sub-graphs can be found in [21]. Furthermore, memory and runtime are challenges for most of the graph mining algorithms [21].

In summary, ILP methods can be useful for learning from structured data if discovery of small substructures is sufficient, but they do require that non-trivial constraints on the search space are provided. If the domain of interest requires the discovery of large substructures, graph mining methods are often more suited. However, sub graph discovery requires calculation of graph isomorphism, which is a NP complete problem. Furthermore, these methods cannot be used to discover several isolated substructures and require exact matching of substructures. Hence, in these cases, more robust methods for learning from structured data are required.

3. Finger Printing

Our approach to structure classification employs a graph transformation method which could address some of the limitations discussed in the previous section. The method does not require a graph isomorphism test and has the ability to combine isolated substructures and has the potential to discover *similar* substructures. It also does not require any constraint to be imposed on the search space. Our method follows a data to model (bottom – up) search strategy and digs down any potential substructures irrespective of its length. Since the graphs are transformed into a canonical form called *finger print*, the computational cost in manipulation of the graphs is very low. Our method could be applied to any form of structured data, from trees to undirected graphs, from sequences to tuples etc., and hence all these types of structured data are referred to as graphs during the rest of this paper.

3.1 The finger printing method

Several methods have been suggested to represent structured data for learning algorithms, and canonical forms of graphs are among the most popular due to their computational simplicity [20]. Our method of transforming structured data into a canonical form of a graph is called *finger printing*.

Structured data is 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 (L_i, L_j, E_k), 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. We refer to such a set as a *finger print*.

The finger prints are used for substructure search in the following way. For all pairs of examples, the intersection of their finger prints, 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 finger prints for which the maximal common substructure is a subset). An upper and lower threshold is applied to select the most contributive substructures for classification. This whole process is solely an item set matching and it successfully avoids the subgraph isomorphism problem. It should be noted that no constraints are applied on the length of the substructures considered during this process. Therefore the discovered substructures are not subjected to pruning the search space beforehand in any manner.

3.2 Implementation

We have developed a feature construction and classifier system called DIFFER (**DI**scovery of **F**eatures using **F**ing**ER** prints), using the methodology described in section 3.1. The input to DIFFER consists of examples of structures that are transformed into graphs. From this, DIFFER produces a set of features together with an encoding of the examples using these features, in a form of a text file that can be used by most standard classification methods (the output file is of the .arff format, which is the recognizable format for WEKA data mining toolkit).

3.3 Illustrative example

As an illustration of how our fingerprinting algorithm works, a toy dataset of 6 molecules is considered. The 2D structures of the molecules are depicted in the 2nd column of Fig. 1 below.

The atom name is considered as the node label for this dataset. The bond types among atoms are the edge (relation) label. Hence the set of distinct node labels for the given dataset is {[N],[C],[O],[S],[Cl]}, and the set of relation labels are {1,2,7}, where 1, 2 and 7 represent single, double and aromatic bonds respectively. Molecules are then transformed into graphs as shown in the 3rd column of the Fig. 1. The transformed graphs could be represented as matrices, where each element of a matrix represents a bond of a certain type between two atoms, as illustrated in the 4th column of Fig. 1.

The fingerprints of the set of examples considered are[1]:

$f(1)$ = {(C,N,1), (O,C,2)}
$f(2)$ = {(C,N, 1), (C,C,7),(O,N,2)}
$f(3)$ = {(C,N,1), (O,C,1), (S,C,1)}
$f(4)$ = {(C,N,1), (C,C,7), (Cl,C,1)}
$f(5)$ = {(C,N,1), (C,C,2), (O,C,1), (Cl,C,1)}
$f(6)$ = {(C,N,1), (O,N,1), (S,N,2)}

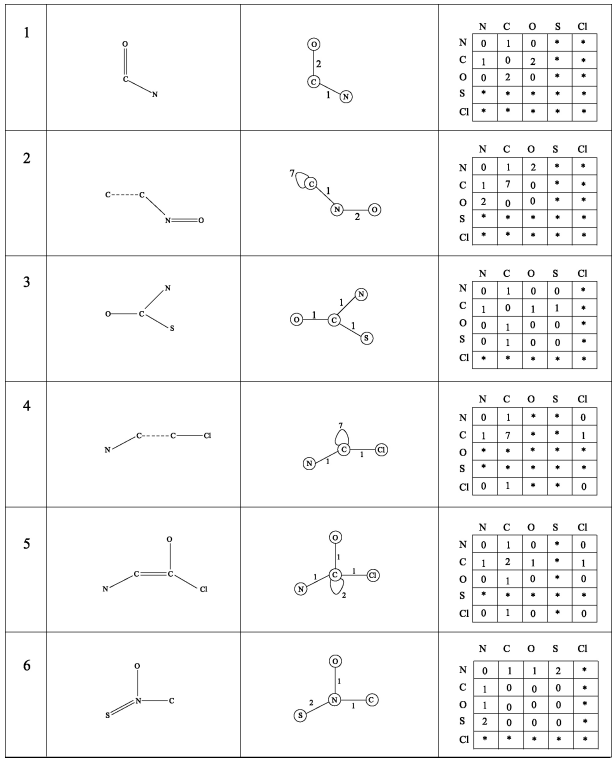

Fig. 1. Example structures and their graph and matrix representations

The maximal common substructure search algorithm is applied to all the fingerprints and the resultant set of substructures selected by the algorithm are: {(C,N,1)}, {(C,N,1), (O,C,1)}, {(C,N,1), (C,C,7)}, {(C,N,1), (Cl,C,1)}. We also apply an upper threshold for removing substructures that appear in more than 95% of the fingerprints. Therefore the selected substructures which could be used as features for an attribute value learner are {(C,C,7)}, {(O,C,1)} and {(Cl,C,1)}, which corresponds to:

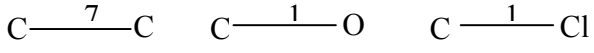

Fig. 2. Selected features from the toy example

This method also allows substructures that are marginally equal, i.e., differ only by few atoms present in the structure, for example carbon chains with different length, would be included in a same feature, enhancing the ability of mining inexact substructures.

[1] Since graphs are undirected, only the lower part of the matrix representation of the graphs (column 4 of Fig. 1.) are considered.

4. Experimental Evaluation

We have used two benchmark datasets from chemo-informatics and one dataset from east-west challenge to compare the performance of DIFFER with other available methods for learning from structures.

4.1 Datasets

First of the three datasets used for experimentation purposes is the mutagenesis dataset [22]. The problem related to the mutagenesis dataset is to predict the mutagenicity *Salmonella typhimurium,* using a set of 230 aromatic and heteroaromatic nitro compounds. Debnath et al [22] has recognized two subsets of this dataset: 188 compounds that could be fitted using linear regression, and 42 compounds that could not. We have used the regression friendly dataset for our evaluation purposes.

The second benchmark data set, carcinogenesis, was originally developed within the US national toxicology program [23]. It consists of 298 compounds that have been shown to be carcinogenic or not in rodents. Although the original dataset contains 3 classes of carcinogenesis, these were treated as one class as done in most previous studies.

The third data set concerns the very popular east-west train problem [24], which contains 20 trains where 10 each are headed to east and west respectively. The task is to identify the characteristics of the trains that make them headed east or west.

4.2 Experimental setup and results

Both Mutagenesis and Carcinogenesis datasets contain atom-bond descriptions of each molecule, as well as the element name and the type of each atom. It also provides the explicit knowledge about complex structures, such as benzene rings, nitro groups etc. Therefore we have considered two levels of background knowledge during the construction of node definitions for the experimentation, such as:

- D 1: atom and bond description of each node is available. Therefore the node definition for D1 is represented as *node(node_name, node_type).* For example a carbon atom of type 22 is defined as *node(c, 22).*
- D 2: Background knowledge includes, in addition to information of D1, the atom's contribution for complex structures such as a part of a benzene ring, or a nitro group, and therefore the respective node definition would be *node(node_name, node_type, [list of structure contributions])*. For example *node(c,22,[N,B,I])*, where N, B and I stands for part of a nitro group, benzene group and a 5-aromatic ring.

The east-west challenge has a set of trains, which each train contains a set of carriages, and a set of loads inside the carriages. These carriages have different properties, such as the number of wheels, roof type etc. and loads has properties such as shape and number. We have considered each train as a structure, and the nodes are the objects it consists of, i.e., *carriages* and *loads*. Node definition in this instant would be *object(object_name,<set of properties>)*. For example a carriage with a long rectangular shape, a flat roof, sides that are not double and 3 wheels, is represented by *object(c,rectangle,long,not_double,flat,3)*. Relations in this domain are *connected_to* and *on*, which has edge labels *1* and *2* respectively.

We have performed experiments with the 3 datasets and feature generation was carried out according to the approach discussed in section 3. We have used all the data as training examples during feature generation. This does not impose any bias on feature construction since we are not considering class distribution of features during the feature construction.

Features generated by DIFFER is used as input to some standard machine learning methods. We have tested for several methods, namely, PART decision list, logistic regression, C4.5 and SVM1 as implemented in WEKA data mining toolkit [26]. DIFFER achieved its best results for the method random forest [25] with 50 trees where 10 random features are evaluated at each node. 10 fold cross validation is used as the evaluation method. The results we obtained with DIFFER were compared with existing state-of-the-art methods, including an ILP based propositionalization method, RSD [16] and graph based propositionalization method for molecules MolFea [15].

We have used all the data in each of the 3 benchmark datasets as training examples for RSD as well, using the same learning methods. RSD also produced its best results for random forest when valuated using 10 fold cross validation. MolFea, which is a specialized tool for mining molecular fragments, has been used for the 1st two datasets and its best results are also included in the Table 1, along with its best classifier within the parenthesis.

We conclude that DIFFER's results are at a par with the existing methods, but the computational simplicity of the method in DIFFER, and ability to address the limitations of the existing methods counts more.

DIFFER's improved accuracy in D2 can be justified as the gain by including molecular topological knowledge into node definition. It also is worth to note that as per to the literature the graph based concept learner SUBDUE-CL [9] have reported a 61.54% accuracy for the carcinogenesis data and the tree mining approach Tree$^2\chi^2$ [11] have reported 80.26% accuracy for the mutagenesis data, which has considerably low performance compared to DIFFER.

Dataset		Accuracy			
		DIFFER	RSD	Molfea	DIFFER + RSD
Mutagenesis	D1	80.61% (RF)	76.6% (RF)	94.7% (Log.)	-
	D2	84.04% (RF)	88.86%[2] (RF)	*[3]	93.76% (RF)
Carcinogenesis	D1	65.25% (RF)	**[4]	67.4% (PART)	-
	D2	68.73% (RF)	54.37% (RF)	*	68.73% (RF)
Trains		80% (RF)	75% (RF)	***[5]	85% (RF)

Table 1. Comparison of DIFFER with some state-of-the-art methods

We also have studied what happens when merging features of DIFFER with those of other methods. When merging the feature set of RSD with the feature set of DIFFER, an increase in accuracy was observed (final column of Table 1). We analyzed the feature set generated by RSD for the mutagenesis dataset and rather surprisingly, we found that it did not contain any atom-bond features. Nonetheless it contained global molecular structure properties such as whether or not two connected nitro groups are present. In contrast to this, the features generated by DIFFER contains inner structural information of atom-bond connections. The experiment demonstrates that by merging these two complementary sets of features the accuracy of the resulting model can be increased.

5. Concluding Remarks

Learning from structured data is an important challenge for machine learning methods, with many important applications, for example within analyzing data from the web, in chemo- and bioinformatics, in management and business transaction domains, etc. These domains are often complex not only in terms of the presence of structures, but also often in terms of the size of the data sets to be analyzed. Existing techniques for learning from structured data are demonstrated to have a number of limitations w.r.t. to effectively analyzing the data due to inability to discover isolated sub-graphs or capture topology of similar sub-graphs and by requiring that non-trivial constraints on the search space is provided, something which may prevent the discovery of large interesting substructures. Standard approaches to graph mining also suffer from the NP-complete subgraph isomorphism problem. In order to overcome these limitations, a novel method, that transforms structured data into a canonical representation, called finger prints, has been presented.

The new method, which has been implemented in a system, called DIFFER, has been shown to be competitive with the existing state-of-the-art methods on some standard benchmark data sets, without imposing constraints on the search space. The reason for its effectiveness can be explained by its ability to mine large as well as isolated discriminative sub-graphs. A very interesting observation is that the classification performance can be improved by merging the features generated by DIFFER with features generated by other methods and thereby integrating the different qualities of several methods. Thus rather than searching for new feature extraction methods that on its own compete with existing methods, it appears to be a promising approach to search for new methods that generate complementary features.

There are several possible directions for future work. At present DIFFER's substructure search is a pair-wise approach, for which the computational cost grows quadratically with the number of examples. A more efficient procedure could be obtained by using some incremental way of searching for the substructures. Sampling of which pairs to consider is also a straightforward way of controlling the computational cost [3]. Alternatives to the use of the covering statistic in conjunction with maximum and minimum thresholds could also be explored. Candidates for this include model driven approaches such as voting by the ROC convex hull or a coverage measure.

The promising result of combining the features generated by DIFFER and RSD also leads to considering merging the features of DIFFER and other methods, perhaps further improving the predictive performance.

[2] We did not achieve the same accuracy for RSD as reported in [16] for the mutagenesis dataset although the same code and files were used in reconstruction of features.

[3] MolFea uses only the atom bond descriptions in constructing its features.

[4] Feature construction algorithm of RSD did not terminate for this data.

[5] MolFea is a molecule fragment miner and cannot be used in any other domain.

References

[1]. M. J. Zaki, C. C. Aggarwal, XRules: an effective structural classifier for XML data, KDD, Washington, USA, 2003, ACM 316–325

[2]. J. R. Quinlan, R. M. Cameron-Jones, FOIL, *Proceedings of the 6th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, Vol. 667*, pp. 3-20. Springer-Verlag 1993

[3]. S. Muggleton, and C. Feng, Efficient induction in logic programs, *Inductive Logic Programming*, pages 281-298. Academic Press, 1992

[4]. A. Srinivasan, R. D. King, and S. Muggleton, "The role of background knowledge: using a problem

from chemistry to examine the performance of an ILP program, *Technical Report PRG-TR-08-99*, Oxford University, 1999.

[5]. S. Muggleton, Inverse entailment and Progol, *New Generation Computing, Special issue on Inductive Logic Programming, 13*(3-4):245-286, 1995

[6]. D. J. Cook, and L. Holder, Substructure discovery using minimum description length and background knowledge, *Journal of Artificial Intelligence Research, 1*:231-255, 1994

[7]. T. Washio and H. Motoda, State of the Art of Graph-based Data Mining, *SIGKDD Explorations Special Issue on Multi-Relational Data Mining*, pp 59-68, Volume 5, Issue 1, 2003

[8]. L. Dehaspe, and H. Toivonen, Discovery of frequent datalog patterns, *Data Mining and Knowledge Discovery, 3*(1):7-36, 1999

[9]. J. Gonzalez, L. B. Holder, and D. J. Cook, Application of Graph-Based Concept Learning to the Predictive Toxicology Domain, *Proceedings of the Predictive Toxicology Challenge Workshop,* 2001

[10]. L. De Raedt, and S. Kramer, The levelwise version space algorithm and its application to molecular fragment finding, *IJCAI'01: Seventeenth International Joint Conference on A. I., volume 2*, pages 853-859, 2001

[11]. B. Bringmann, and A. Zimmermann, Tree - Decision Trees for Tree Structured Data, *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005), Notes in Artificial Intelligence, (LNAI) 3721*, pp. 46-58, Springer 2005

[12]. T. Horváth, T. Gärtner, and S. Wrobel, Cyclic pattern kernels for predictive graph mining, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* Pages: 158 - 167 Seattle, WA, USA, 2004

[13]. J. Ramon, and T. Gaertner, Expressivity versus efficiency of graph kernels, *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65-74 ,2003

[14]. K. M. Borgwardt, and H. P. Kriegel, Shortest-Path Kernels on Graphs, *ICDM* 2005, pp: 74-81

[15]. C. Helma, S. Kramer, and L. De Raedt, The Molecular Feature Miner Molfea, "Molecular Informatics: Confronting Complexity", *Proceedings of the Beilstein-Institut Workshop,* Bozen, Italy, 2002

[16]. M-A. Krogel, S. Rawles, F. Železný, P. A. Flach, N. Lavrač, and S. Wrobel, Comparative evaluation of approaches to propositionalization, *Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'2003), number 2835* in Lecture Notes in Computer Science, pages 197--214, Springer Verlag 2003

[17]. N. Lavrac, and P. Flach, An extended transformation approach to Inductive Logic Programming, *University publication*, Department of Computer science, University of Bristol, 2000

[18]. N. Lavrac, F. Zelezny, and P. Flach, RSD: Relational Subgroup Discovery through First-order Feature Construction, *Proceedings of the 12th International Conference on Inductive Logic Programming (ILP'02)*, Springer-Verlag, ISBN 3-540-00567-6, 2002

[19]. C. Nattee, S. Sinthupinyo, M. Numao, T. Okada, Inductive Logic Programming for Structure-Activity Relationship Studies on Large Scale Data, *SAINT Workshops 2005*: 332-335, 2005

[20]. A. Inokuchi, T. Washio, and H. Motoda, Complete mining of frequent patterns from graphs, *Mining graph data, Machine Learning, 50*:321-354, 2003

[21]. I. Fischer, and T. Meinl, Graph based molecular data mining - an overview, *IEEE SMC 2004 Conference Proceedings*, pages 4578—4582, 2004

[22]. A. K. Debnath, Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., and C. Hansch, Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds: Correlation with molecular orbital energies and hydrophobicity, Journal Med. Chem. 34:786-797

[23]. US National Toxicology program, http://ntp.niehs.nih.gov/index.cfm?objectid=32BA9724-F1F6-975E-7FCE50709CB4C932

[24]. D. Michie, S. Muggleton, D. Page, and A. Srinivasan,, To the international computing community: A new East-West challenge, *Oxford University Computing laboratory*, Oxford, UK, 1994, ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/trains.tar.Z

[25]. L. Breiman, Random Forests, *Machine Learning 45*(1): 5-32, 2001

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