Expert Systems with Applications 42 (2015) 6442–6456

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Wordification: Propositionalization by unfolding relational data into bags of words

Matic Perovšek [a,b,*], Anže Vavpetič [a,b], Janez Kranjc [a,b], Bojan Cestnik [a,c], Nada Lavrač [a,b,d]

[a] Jožef Stefan Institute, Ljubljana, Slovenia
[b] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[c] Temida d.o.o, Ljubljana, Slovenia
[d] University of Nova Gorica, Nova Gorica, Slovenia

ARTICLE INFO

Article history:
Available online 24 April 2015

Keywords:
Wordification
Inductive Logic Programming
Relational Data Mining
Propositionalization
Text mining
Classification

ABSTRACT

Inductive Logic Programming (ILP) and Relational Data Mining (RDM) address the task of inducing models or patterns from multi-relational data. One of the established approaches to RDM is propositionalization, characterized by transforming a relational database into a single-table representation. This paper presents a propositionalization technique called *wordification* which can be seen as a transformation of a relational database into a corpus of text documents. Wordification constructs simple, easy to understand features, acting as words in the transformed Bag-Of-Words representation. This paper presents the wordification methodology, together with an experimental comparison of several propositionalization approaches on seven relational datasets. The main advantages of the approach are: simple implementation, accuracy comparable to competitive methods, and greater scalability, as it performs several times faster on all experimental databases. Furthermore, the wordification methodology and the evaluation procedure are implemented as executable workflows in the web-based data mining platform ClowdFlows. The implemented workflows include also several other ILP and RDM algorithms, as well as the utility components that were added to the platform to enable access to these techniques to a wider research audience.

1. Introduction

Standard propositional data mining algorithms, included in established data mining tools like Weka (Witten, Frank, & Hall, 2011), induce models or patterns learned from a single data table. On the other hand, the aim of Inductive Logic Programming (ILP) and Relational Data Mining (RDM) is to induce models or patterns from multi-relational data (De Raedt, 2008; Džeroski & Lavrač, 2001; Lavrač & Džeroski, 1994; Muggleton, 1992). Most types of propositional models and patterns have corresponding relational counterparts, such as relational classification rules, relational regression trees or relational association rules.

For multi-relational databases in which data instances are clearly identifiable (the so-called individual-centered representation (Flach & Lachiche, 1999), characterized by one-to-many relationships among the target table and other data tables), various techniques can be used for transforming a multi-relational database into a propositional single-table format (Krogel et al., 2003). After performing such a transformation (Lavrač, Džeroski, & Grobelnik, 1991), named *propositionalization* (Kramer, Pfahringer, & Helma, 1998), standard propositional learners can be used, including decision tree and classification rule learners.

Inspired by text mining, this paper presents a propositionalization approach to Relational Data Mining, called *wordification*. Unlike other propositionalization techniques (Kramer et al., 1998; Kuželka & Železný, 2011; Lavrač et al., 1991; Železný & Lavrač, 2006), which first construct complex relational features (constructed as a chain of joins of one or more tables related to the target table), used as attributes in the resulting tabular data representation, wordification generates much simpler features with the aim of achieving greater scalability.

Wordification can be viewed as a transformation of a relational database into a set of feature vectors, where each original instance is transformed into a-kind-of 'document' represented as a Bag-Of-Words (BOW) vector of weights of simple features, which can be interpreted as 'words' in the transformed BOW space. The 'words' constructed by wordification correspond to individual

* Corresponding author at: Jožef Stefan Institute, Ljubljana, Slovenia.
E-mail addresses: matic.perovsek@ijs.si (M. Perovšek), anze.vavpetic@ijs.si (A. Vavpetič), janez.kranjc@ijs.si (J. Kranjc), bojan.cestnik@temida.si (B. Cestnik), nada.lavrac@ijs.si (N. Lavrač).

http://dx.doi.org/10.1016/j.eswa.2015.04.017
0957-4174/

attribute–values of the target table and of the related tables, subsequently weighted by their Term Frequency-Inverse Document Frequency (TF-IDF) value (Jones, 1972; Salton & Buckley, 1988) (requiring real-valued attributes to be discretized first). Alternatively, instead of TF-IDF, simpler schemes can be used such as term frequency (TF) 'word' count, or the binary scheme indicating just the presence/absence of a 'word' in the 'document'.

To intuitively phrase the main idea of wordification, take two simple examples illustrating the wordification data preprocessing step in class-labeled data, where each structured data instance is transformed into a tuple of simple features, which are counts/weights of individual attribute–value pairs. Take the well-known relational domain of East–West Trains (Michie, Muggleton, Page, & Srinivasan, 1994) with cars containing different loads: one of the train's features in the BOW representation is the count/weight of rectangular loads it carries, no matter in which cars these loads are stored. Or in the standard Mutagenesis domain (Srinivasan, Muggleton, King, & Sternberg, 1994), a molecule may prove to be toxic if it contains a lot of atoms characterized by the property *atom_type = lead*, no matter how these atoms are bonded in the molecule. The main hypothesis of the wordification approach is that the use of this simple representation bias is suitable for achieving good results in classification tasks. Moreover, when using a binary scheme, this representation bias allows for simple and very intuitive interpretation in descriptive induction tasks, such as association rule learning from unlabeled multi-relational data.

Wordification suffers from some loss of information, compared to propositionalization methods which construct complex first-order features (which get values *true* or *false* for a given individual) as a chain of joins of one or more tables related to the target table. Nevertheless, despite some information loss, wordification has numerous advantages. Due to the simplicity of features, the generated hypotheses are easily interpretable by domain experts. The feature construction step in wordification is very efficient, therefore it can scale well for large relational databases. As wordification constructs each 'document' independently from the other 'documents', a large main table can be divided into smaller batches of examples, which can be propositionalized in parallel. Next, wordification can use TF or TF-IDF word weighting to capture the importance of a given feature (attribute value) of a relation in an aggregate manner, while feature dependence is modeled by constructing a-kind-of word '*n*-grams' as conjuncts of a predefined number of simple features. Finally, the wordification approach has the advantage of using techniques developed in the text mining community, such as efficient document clustering or word cloud visualization, which can now be effectively exploited in multi-relational data mining.

This paper shows that the developed wordification technique is simple, considerably more efficient and at least as accurate as the comparable state-of-the-art propositionalization methods. This paper extends our previous research (Perovšek, Vavpetič, & Lavrač, 2012, 2013) in many ways. The related work is more extensively covered. The improvements to the methodology include feature filtering by frequency, performance optimization (indexing by value), new options regarding feature weighting (next to TF-IDF, we added TF and binary), and a parallel version of the algorithm. The methodology description is now more detailed, including the formal wordification framework, the wordification algorithm pseudo code as well as time and space complexity analysis. The experimental evaluation has been substantially extended to include a comparison of three different term weighting schemes, additional propositionalization algorithms RelF (Kuželka & Železný, 2011) and Aleph (Srinivasan, 2007), as well as an additional classifier (SVM), which were applied to an extended set of experimental relational datasets. Such exhaustive experimentation has enabled us to statistically validate the experimental results by using the Friedman test and the Nemenyi post hoc test on the seven benchmark problems from the five relational domains (two of which have two database variants): IMDB,[1] Carcinogenesis (Srinivasan, King, Muggleton, & Sternberg, 1997), Financial[2] and two variants of Trains (Michie et al., 1994) and Mutagenesis (Srinivasan et al., 1994). Further experiments were done to analyze the effects of feature weighting, pruning and *n*-gram construction. In addition to the two experimental workflows developed in the web-based data mining platform ClowdFlows (Kranjc, Podpečan, & Lavrač, 2012), one workflow developed for learning and another for results evaluation and visualization, this paper introduces another wordification workflow applicable in association rule learning tasks from binarized features. The implemented workflows, which are available online through ClowdFlows, allow for methodology reuse and experiment repeatability. As a side-product of workflow development, the competing propositionalization algorithms used in experimental comparisons are also made available through ClowdFlows and can therefore be easily reused in combination with numerous pre-existing ClowdFlows components for data discretization, learning, visualization and evaluation, including a large number of Weka (Witten et al., 2011) and Orange (Demšar, Zupan, Leban, & Curk, 2004) components. Making selected RDM algorithms handy to use in real-life data analytics may therefore contribute to improved accessibility and popularity of Relational Data Mining.

The paper is organized as follows. Section 2 describes the background and the related work. Section 3 gives an informal overview of the wordification methodology, while Section 4 presents the formalism and the details of the developed wordification algorithm. The implementation of the methodology as a workflow in the ClowdFlows platform is described in Section 5. Section 6 presents the evaluation methodology implementation and the experimental results. Section 7 illustrates the utility of wordification in a descriptive induction setting of learning association rules from two real-life domains, using data from a subset of the IMDB movies database and from a database of traffic accidents. Section 8 concludes the paper by presenting the plans for further work.

2. Background and related work

Inductive Logic Programming (ILP) and Relational Data Mining (RDM) algorithms are characterized by the ability to use background knowledge in learning relational models or patterns (Džeroski & Lavrač, 2001; De Raedt, 2008; Lavrač & Džeroski, 1994; Muggleton, 1992), as by taking into account additional relations among the data objects the performance of data mining algorithms can be significantly improved.

Propositionalization (Kramer et al., 1998; Lavrač et al., 1991) is an approach to ILP and RDM, which offers a way to transform a relational database into a propositional single-table format. In contrast to methods that directly induce relational patterns or models, such as Aleph (Srinivasan, 2007) and Progol (Muggleton, 1995), propositionalization algorithms transform a relational problem into a form which can be solved by standard machine learning or data mining algorithms. Consequently, learning with propositionalization techniques is divided into two self-contained phases: (1) relational data transformation into a single-table data format and (2) selecting and applying a propositional learner on the transformed data table. As an advantage, propositionalization is not limited to specific data mining tasks such as classification, which is usually the case with ILP and RDM methods that directly induce models from relational data.

[1] http://www.webstepbook.com/supplements/databases/imdb.sql.
[2] http://lisp.vse.cz/pkdd99/Challenge/berka.htm.