Knowledge-Based Systems 23 (2010) 302–308

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Text classification using graph mining-based feature extraction

Chuntao Jiang *, Frans Coenen, Robert Sanderson, Michele Zito

The University of Liverpool, Department of Computer Science, Ashton Building, Ashton Street, Liverpool, L69 3BX, United Kingdom

ARTICLE INFO

Article history:
Available online 22 November 2009

Keywords:
Text classification
Graph representation
Graph mining
Weighted graph mining
Feature extraction

ABSTRACT

A graph-based approach to document classification is described in this paper. The graph representation offers the advantage that it allows for a much more expressive document encoding than the more standard bag of words/phrases approach, and consequently gives an improved classification accuracy. Document sets are represented as graph sets to which a weighted graph mining algorithm is applied to extract frequent subgraphs, which are then further processed to produce feature vectors (one per document) for classification. Weighted subgraph mining is used to ensure classification effectiveness and computational efficiency; only the most significant subgraphs are extracted. The approach is validated and evaluated using several popular classification algorithms together with a real world textual data set. The results demonstrate that the approach can outperform existing text classification algorithms on some dataset. When the size of dataset increased, further processing on extracted frequent features is essential.

1. Introduction

The most common document formalisation for text classification is the *vector space* model founded on the bag of words/phrases representation. The main advantage of the vector space model is that it can readily be employed by classification algorithms. However, the bag of words/phrases representation is suited to capturing only word/phrase frequency; structural and semantic information is ignored. It has been established that structural information plays an important role in classification accuracy [14].

An alternative to the bag of words/phrases representation is a graph based representation, which intuitively possesses much more expressive power. However, this representation introduces an additional level of complexity in that the calculation of the similarity between two graphs is significantly more computationally expensive than between two vectors (see for example [16]). Some work (see for example [12]) has been done on hybrid representations to capture both structural elements (using the graph model) and significant features using the vector model. However the computational resources required to process this hybrid model are still extensive.

The computational complexity of the graph representation for text mining is the main disadvantage of the approach, which prevents the full exploitation of the expressive power that the graph representation possesses. The work described in this paper seeks to address this issue by applying weighted graph mining analysis to the problem. The intuition behind the approach is that in standard frequent subgraph mining all generated subgraphs are assumed to have equal importance. However it is clear that, at least in the context of text mining, some subgraphs are more significant than others.

The rest of this paper is organized as follows. In Section 2 a brief overview of previous work is presented. The graph representation of document sets is then discussed in Section 3. In Section 4 the weighted subgraph mining is defined. The proposed weighted graph mining algorithm, a variation of gSpan called Weighted gSpan (W-gSpan), is introduced in Section 5. A set of evaluating experiments are then presented in Section 6, followed by some concluding remarks in Section 7.

2. Related work

Much early work on document graph representations for text classification was directed at Web documents. Geibel et al. in [7] demonstrated that it is possible to classify Web documents using document structure alone; however we shall demonstrate that a much more powerful approach is to combine structure with linguistic and semantic information. For example Schenker [16] proposed a number of methods to represent Web documents as graphs so as to include the structural information of the Web documents. The typical approach is to conduct classification using some similarity-based algorithm. However, approaches that operate using graph similarity measures are computationally expensive (for example computing the "maximum common subgraph" between two graphs is a NP hard problem [5]). Hybrid representations have been introduced to resolve the computational overhead associated with pure graph representations, see for example [12]. Such hybrid representations are reported to have

* Corresponding author. Tel.: +44 0151 7954275; fax: +44 0151 7954235.
E-mail addresses: c.jiang@liv.ac.uk (C. Jiang), coenen@liv.ac.uk (F. Coenen), azaroth@liv.ac.uk (R. Sanderson), michele@liv.ac.uk (M. Zito).

0950-7051/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2009.11.010

better performance than pure graph based methods. However the computational resources required to process these hybrid model are still very high due to: (i) the extremely high number of nodes and edges, low number of edge labels and high repetition of structural node labels, encountered; and (ii) the consequent exponential complexity of the search space.

The use of graphs for representing text has a very long history in Natural Language Processing (NLP). However the work in NLP has focused on language understanding techniques such as Part Of Speech (POS) tagging, rather than text classification. Previous work [13,20] has looked at the collocation of terms and their frequencies as graphs, rather than the linguistic structure of the sentence. One other study [6] has represented linguistic information as well as word order in a graph for text classification, however the work was limited to very small texts of between 8 and 13 tokens such as the titles of works. As such, we adopt the usage of linguistic information, structure and semantics in a graph for text classification at a full text scale. In order to achieve this scale of processing, the use of frequent subgraph mining is essential.

Frequent subgraph (and sub-tree) mining, using various approaches, has been extensively studied [9,10,22,8,2]. However, the main bottleneck is the number of unnecessary candidate frequent subgraphs generated. A substantial amount of work has been undertaken focusing on developing efficient graph mining algorithms using elegant search strategies, data structures or their combinations. Some authors have suggested the use of constraint based frequent subgraph mining to remove unwanted patterns. The weighted subgraph mining approach advocated in this paper integrates the weight constraints into the frequent subgraph mining process to reduce the search space by generating only the most significant (interesting) patterns.

The frequent subgraph mining approach described in this paper is also influenced by work on weighted pattern mining, especially Weighted Association Rules Mining (WARM), see for example the work of [19,17,23–25]. A significant issue in WARM is that the "Downward Closure" (DC) property of items sets, on which many ARM algorithms are based, no longer holds. One solution (for example [19]) is to handle the weights as a post-processing step after mining frequent itemsets, however the weights are then not integrated into the ARM process. Tao et al. [17] proposed a model of weighted support, which satisfies a weighted DC property. Yun et al. [23–25] introduced a series of concepts such as "weight range", "weight confidence", and "support confidence" for WARM in order to maintain the DC property and push the weight constraint deeply into the mining process. Although the ideas espoused by WARM cannot be directly applied to weighted frequent subgraph mining; the research described here is, at least in part, influenced by this body of work.

3. Graph representation of text data

The graph representation advocated in this paper is described in this section. The representation serves to capture a range documents aspects: (i) word stem, (ii) word Part Of Speech (POS), (iii) word order, (iv) word hypernyms, (v) sentence structure, (vi) sentence division and (vii) sentence order. There are four different types of nodes in the graph representation:

1. *Structural*: Nodes that represent sentences (S) and their internal structures of noun (NP), verb (VP) and prepositional phrases (PP). (Represented by triangles in Fig. 1.)
2. *Part of Speech*: Nodes that represent the POS of a word, (e.g. DT, JJ, and NN). (Circles.)
3. *Token*: Nodes that represent the actual word tokens in the text. (Rectangles.)
4. *Semantic*: Nodes that represent additional information about the word such as its linguistic stem and other broader concepts. (Ovals)

Note that each node has a unique identifier and a label. There are also five types of edge in the graph:

1. *hasChild*: Edges which record the structure of the text such as a sentence having a noun phrase and a verb phrase or a noun phrase containing an adjective. (Unlabeled in Fig. 1 for reasons of space.)
2. *isToken*: Edges which link the part of speech of a token to the token itself.
3. *next*: Edges which record the order of the words and sentences in the text.
4. *stem*: Edges which link to the linguistic stem of the word.
5. *hyp*: Edges which link to a broader concept.

An example of these node and edge types is depicted in Fig. 1, using the first 6 words in a well known English sentence. Employing the above graph representation each sentence in each text is encoded and linked together with "next" edges to form one graph per text. Content based weightings were then attached to each node in the graph. The Structural elements, being intuitively unimportant to classification, were given a static low weight of 1. The Part of Speech nodes were given a static weight of 10, Token nodes were weighted according to their frequency in the dataset using the $TF \cdot IDF$ method. Stems were half the value of the Token and Hypernyms one quarter the value.

4. Weighted frequent subgraphs

In this section the weighted subgraph mining problem is formally defined. As with standard transaction graph mining approaches [9,10,1,11] we commence with a set of *transaction graphs* $D = \{G_1, G_2, \ldots, G_n\}$ and a function $\tau(g, G)$ for arbitrary graphs g and G. $\tau(g, G) = 1$ (resp. 0), if g is isomorphic to a subgraph in G.

Definition 1. The support count of a graph (pattern) g with respect to a database $D = \{G_1, G_2, \ldots, G_n\}$, is the expression $sco(g) = \sum_{i=1}^{i=n} \tau(g, G_i)$. The support of g with respect to D, $sup(g)$, is the ratio of the support count over the size of the dataset D. Then:

$$sup(g) = \frac{sco(g)}{n}. \quad (1)$$

It should be remarked that $sco(g)$ and $sup(g)$, like most terms defined in this section depend on the dataset D. To avoid cluttering notations, such dependence will always be left implicit.

Definition 2. Given a graph g, if $sup(g)$ is greater than or equal to some user defined minimum threshold θ, then g is said to be frequent (in D). The frequent subgraph mining problem is to find all the frequent subgraphs in the database D.

Since the purpose of this paper is to study weighted graph mining in the remainder of this section we define this concept precisely. From now on we assume that graphs come with weights associated with either their vertices or their edges. Let W be a function assigning a weight to any graph g in terms of the given weights for its vertices (resp. edges). In our work, in particular, W will always be a sum of the vertex (resp. edge) weights, but the definitions in this section hold in a more general setting.

Definition 3. Given a graph g with the weight $W(g)$, the weighted support of g with respect to D, $wsup(g)$, is:

$$wsup(g) = W(g) \times sup(g). \quad (2)$$