Pattern Recognition Letters 85 (2017) 29–34

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Sentence level matrix representation for document spectral clustering

Víctor Mijangos[a,*], Gerardo Sierra[a], Azucena Montes[b]

[a] Language Engineering Group, National Autonomous University of Mexico, Mexico D.F. 04510, Mexico
[b] National Center for Research and Technological Development, Cuernavaca 62490, Mexico

ARTICLE INFO

Article history:
Received 9 March 2016
Available online 21 November 2016

MSC:
41A05
41A10
65D05
65D17

Keywords:
Natural language analysis
Information retrieval
Graph-based clustering
Document processing
Spectral clustering
Frobenius metric
Doc2Vec

ABSTRACT

Using a simple vector in $\mathbb{R}^n$ is a traditional way of representing documents in vector spaces. However, this representation tends to ignore the discourse and syntactic structure of texts. A matrix representation such as the one offered by the Doc2Vec word embedding method preserves these characteristics. In order to integrate a sentence level matrix representing documents to a clustering algorithm, we use a Frobenius based inner product that allows defining kernel functions for spectral clustering. We show that this methodology provides advantages over traditional clustering algorithms and performs better than bag of words (BoW) representations used in Information Retrieval (IR).

1. Introduction

Nowadays, the amount of information generated through several sources, including media, is enormous. Machine learning algorithms try to solve the problem of clustering this information into particular groups in order to try to automatically classify it. When this data is coded into natural language, another problem arises: How to represent natural language for clustering? There are many ways. A traditional way of representing documents into vector spaces is by using a simple vector in $\mathbb{R}^n$. However, this way of representation tends to ignore the discourse and syntactic structure of texts. In contrast, a matrix representation such as the one offered by the Doc2Vec word embedding method preserves these characteristics. In addition, word embedding has gained relevance in the Natural Language Processing community because of this methods high performance in different tasks such as automatic summarization, sentiment analysis, and word clustering, among others. Nevertheless, currently, no application of word embedding techniques for document spectral clustering exists.

In this paper, we propose clustering documents based on sentence level matrix representations through the use of spectral relaxation, considering that it is a good technique to make a linearly separable space when data was not in the original space. We propose a set of Frobenius-based kernel functions for the purpose of applying such spectral relaxation to the sentence level matrix representation of the documents.

First, we show previous work done in the area of document representations and clustering. In Section 3 we describe the methodology based on Doc2Vec representations of documents, on spectral relaxation and on the *k*-means algorithm. Next, in Section 4 we show the application of the methodology to the Corpus of Sexualities of Mexico (CSMX) and the evaluation of this proposal. Finally the conclusions are presented in Section 5.

2. Previous work

In this section we address previous works on document representations, focusing on the Distributional Space Models (DSMs). Next, we show the clustering problem, providing a definition of what clustering is and summarizing the different methods.

2.1. Document representation

In Information Retrieval, a common way of representing documents is through bag of words (BoW). For this, the idea is to create vectors of documents, that is, to represent them by a vector filled with the weights of the words in the document. To assign

* Corresponding author.
E-mail address: vmijangosc@iingen.unam.mx (V. Mijangos).

http://dx.doi.org/10.1016/j.patrec.2016.11.008
0167-8655/

the weight to each word, the most common method is the Tfldf formula (Eqs. (1)–(3)) [23]. Let w be a word in a document d. Let D be the corpus containing such documents, and let $f(w)$ be the frequency of the word in document d; then we can define Tfldf as follows:

$$tf = 0.5 + \frac{0.5 \cdot f(w)}{max\{f(w_i) : w_i \in d\}} \tag{1}$$

$$idf = log\frac{|D|}{|\{d \in D : w \in d\}|} \tag{2}$$

$$tfidf = tf \cdot idf \tag{3}$$

However, this kind of representation fails to capture some important features of the natural languages, like the distribution of words between the documents (words can be represented as repeated weights, even if it is not the exact same word). This method only captures features on the lexical level. Thus, different types of representations have replaced the bag of words method. Currently, distributional methods have had a great acceptance and they are the state of the art for many tasks [6,32]. These methods are based on the ideas of Harris [13], who said that there are distributional regularities between different items of natural language. Therefore, Sahlgren [33] says that words occurring in similar contexts have similar meanings.

Based on the previous ideas, distributional ways of representing words have emerged recently. Distributional Space Models (DSM) have shown better performance than the simple bag of words method. One example of these methods is the Latent Semantic Analysis [19]. This analysis consists on building a matrix of co-occurrences of a word in different contexts (like documents); then the matrix is decomposed into three other matrices by Singular Value Decomposition (SVD) and is composed again taking the first k relevant terms. The idea behind this is that the composed matrix lacks noise and represents better the semantic elements.

Other methods similar to LSA are the Probabilistic Latent Semantic Analysis (PLSA) [14] and the Latent Dirichlet Allocation (LDA) [5]. Also, among the DSMs, there are other proposed models like Hyperspace Analogue to Language (HAL), BEAGLE, random indexing, and others [6].

In general, Baroni et al. [1] proposed a program on distributional models and compositional semantics. The main idea is that the vectors of the words compose the vector of a phrase or a sentence. Three ways are proposed to do this compositions: (a) sum of vectors; (b) product between vectors; and (c) by learning matrix decomposition (see [12]).

Finally, Bengio et al.[4] proposed a stochastic neural model based on Soft-Max regression. In this way, a vector space based on this model was presented in [26]. This embedding method performed better than the regular DSMs as was shown by Baroni et al. [2]. This is what we use here.

2.2. Clustering

The main problem of a clustering procedure is that we have a limited training set $X = \{x_i : x_i \in \mathbb{R}^n, i = 1, \ldots, m\}$ and we need to determinate k clusters separating this set based on well-established criteria [34]. According with [36] clustering can be defined as follows:

Definition 1 (Clustering)**.** Given a set of objects X, a clustering $\mathfrak{C} = \{C_i : C_i \subseteq X, i = 1, \ldots, k\}$ is a partition of X such that

1. $\bigcup_{C_i \in \mathfrak{C}} C_i = V$
2. $\forall C_i, C_j \in \mathfrak{C} : C_i \cap C_j = \emptyset$ for $i \neq j$

In the case of documents, the main criteria, and which we adopted here, are the thematic relations. The partitional and hierarchical algorithms have been used for this task [34]. Currently, the fuzzy methods have raised new views [15,21]. However the k-means algorithm (part of the partitional methods) is characterized to be one of the most computationally efficient clustering algorithms, achieving a complexity of $O(Nkn)$ [38]. However, as pointed out by Rui Xu et al. [38], the fact that the k-means algorithm does not tackle high dimensional data is worth nothing.

Regarding Natural Language Processing (as well as other areas) we must deal with the *curse of dimensionality* in the vector representation of data. For this reason, it is common to reduce the dimensionality of the vectors (for example with Principal Component Analysis (PCA) or Singular Value Decomposition (SVD)). In addition, the use of the methodology of spectral relaxation described in [3,10,30,37], among others is useful for dimensionality reduction.

This view has fostered the use of the spectral clustering algorithm. As Luxburg [37] points out, this algorithm has become one of the most popular clustering algorithms and a technique largely used for image clustering. According to this perspective, the advantage of this technique is that it permits a high reduction of data, while preserving its natural groups. This view will be addressed in Section 3.2.

3. Methodology description

The methodology of the procedure proposed in this article is based on three major modules: (i) the Doc2Vec method for representing a document in a vector space; (ii) spectral relaxation as a way of reducing the problem to the partition of a graph; and (iii) the k-means algorithm for the clustering. Below we will explain all three tools and show the advantages of each.

3.1. Representation of documents in vector space

The traditional BoW representation of documents through Tfldf has been largely overpassed in different tasks by new methods like Doc2Vec representations [7]. As shown by Campr and Ježek [7], the Doc2Vec representation of documents has better performance than different vector space representation procedures.

Doc2Vec modifies the Word2Vec model proposed by Mikolov et al. [26]. Doc2Vec is based on the Skip-gram model [27], which has been parameterizing using a Soft-Max regression:

$$P(c|w;\theta) = \frac{e^{v_c v_w}}{\sum_{c^i \in C} e^{v_{c^i} v_w}} \tag{4}$$

where C is the set of contexts such that $c \in C$, w is a word and $v_c, v_w \in \mathbb{R}^n$ are vector representations of c and w respectively. The goal of Word2Vec is to represent the words as points in space by its continuous embedding. It begins with random space points and aims to maximize Eq. (4) while minimizing the loss function $E = -logP(v_w|v_c)$.

As [26] suggests, the negative-sampling approach is the more efficient way of obtaining word embedding. This way, Eq. (4) takes the form:

$$P(D = 1|w, c;\theta) = \frac{1}{1 + e^{-v_c v_w}} \tag{5}$$

where $D = 1$ implies that w and c came from the corpus data. So the Word2Vec (as well as the Doc2Vec) algorithm seeks to maximize the equation:

$$log\sigma(v_c v_w) + \sum_{i=1}^{m} \mathbb{E}_{w_i \sim P_n(w)}[log\sigma(-v_c v_w)] \tag{6}$$

where σ equals Eq. (5). As [20] says, the Doc2Vec algorithm is based on the ideas of Word2Vec, namely on Eq. (6). However,