Pattern Recognition Letters 42 (2014) 79–84

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Authorship analysis based on data compression ☆

Daniele Cerra *, Mihai Datcu, Peter Reinartz

German Aerospace Center (DLR), Muenchner str. 20, 82234 Wessling, Germany

ARTICLE INFO

Article history:
Received 13 May 2013
Available online 6 February 2014

Keywords:
Authorship analysis
Data compression
Similarity measure

ABSTRACT

This paper proposes to perform authorship analysis using the Fast Compression Distance (FCD), a similarity measure based on compression with dictionaries directly extracted from the written texts. The FCD computes a similarity between two documents through an effective binary search on the intersection set between the two related dictionaries. In the reported experiments the proposed method is applied to documents which are heterogeneous in style, written in five different languages and coming from different historical periods. Results are comparable to the state of the art and outperform traditional compression-based methods.

1. Introduction

The task of automatically recognizing the author of a given text finds several uses in practical applications, ranging from authorship attribution to plagiarism detection, and it is a challenging one [33]. While the structure of a document can be easily interpreted by a machine, the description of the style of each author is in general subjective, and therefore hard to derive in natural language; it is even harder to find a description which enables a machine to automatically tell one author from the other. A literature review on modern authorship attribution methods, usually coming from the fields of machine learning and statistical analysis, is reported in [33,16,21,14,18]. Among these, algorithms based on similarity measures such as [3,22] are widely employed and usually assign an anonymous text to the author of the most similar document in the training data.

During the last decade, compression-based distance measures have been effectively applied to cluster texts written by different authors [10] and to perform plagiarism detection [7]. Such universal similarity measures, of which the most well-known is the Normalized Compression Distance (NCD), employ general compressors to estimate the amount of shared information between two objects. Similar concepts are also used by methods using runlength histograms to retrieve and classify documents [13]. Experiments carried out in [27] conclude that NCD-based methods for authorship analysis outperform state-of-the-art classification methodologies such as Support Vector Machines. A study on larger and more statistically meaningful datasets shows NCD-methods to be competitive with respect to the state of the art [12], while [33] reports that compression-based methods are effective but hard to use in practice as they are very slow.

Indeed the universality of these measures comes at a price, as the compression algorithm must be run at least n^2 times on n objects to derive a distance matrix, slowing down the analysis. Furthermore, as these methods are applied to raw data they cannot be tuned to increase their performance on a given data type. We propose then to perform these tasks using the Fast Compression Distance (FCD) recently defined in [6], which provides superior performances with a reduced computational complexity with respect to the NCD, and can be tuned according to the kind of data at hand. In the case of natural texts, only FCD's general settings should be adjusted according to the language of the dataset, thus keeping the desirable parameter-free approach typical of NCD. Applications to authorship and plagiarism analysis are derived by extracting meaningful dictionaries directly from the strings representing the data instances and matching them. The reported experiments show that improvements over traditional compression-based analysis can be dramatic, and that the FCD could become an important tool of easy usage for the automated analysis of texts, as satisfactory results are achieved skipping any parameters setting step. The only exception is an optional text preprocessing step which only needs to be set once for documents of a given language, and does not depend on the specific dataset.

The paper is structured as follows. Section 2 introduces compression-based similarity measures and the FCD, which will be validated in an array of experiments reported in Section 3. We conclude in Section 4.

☆ This paper has been recommended for acceptance by G. Moser.

* Corresponding author. Tel.: +49 8153 28 1496; fax: +49 8153 28 1444.
E-mail address: daniele.cerra@dlr.de (D. Cerra).

http://dx.doi.org/10.1016/j.patrec.2014.01.019
0167-8655/© 2014 Elsevier B.V. All rights reserved.

2. Fast Compression Distance

Compression-based similarity measures exploit general off-the-shelf compressors to estimate the amount of information shared by any two objects. They have been employed for clustering and classification on diverse data types such as texts and images [35], with [19] reporting that they outperform general distance measures. The most widely known and used of such notions is the Normalized Compression Distance (NCD), defined for any two objects x and y as:

$$NCD(x,y) = \frac{C(x,y) - \min C(x), C(y)}{\max C(x), C(y)} \quad (1)$$

where $C(x)$ represents the size of x after being compressed by a compressor (such as Gzip), and $C(x,y)$ is the size of the compressed version of x appended to y. If $x = y$, the NCD is approximately 0, as the full string y can be described in terms of previous strings found in x; if x and y share no common information the NCD is $1 + e$, where e is a small quantity (usually $e < 0.1$) due to imperfections characterizing real compressors. The idea is that if x and y share common information they will compress better together than separately, as the compressor will be able to reuse recurring patterns found in one of them to more efficiently compress the other. The generality of NCD allows applying it to diverse datatypes, including natural texts. Applications to authorship categorization have been presented by Cilibrasi and Vitányi [10], while plagiarism detection of students assignments has been succesfully carried out by Chen et al. [7].

A modified version of NCD based on the extraction of dictionaries has been first defined by Macedonas et al. [24]. The advantages of using dictionary-based methods have been then studied by Cerra and Datcu [6], in which the authors define a Fast Compression Distance (FCD), and succesfully apply it to image analysis. The algorithm can be used for texts analysis as follows.

First of all, all special characters such as punctuation marks are removed from a string x, which is subsequently tokenized in a set of words W_x. The sequence of tokens is analysed by the encoding algorithm of the Lempel–Ziv–Welch (LZW) compressor [36], with the difference that words rather than characters are taken into account. The algorithm initializes the dictionary $D(x)$ with all the words W_x. Then the string x is scanned for successively longer sequences of words in $D(x)$ until a mismatch in $D(x)$ takes place; at this point the code for the longest pattern p in the dictionary is sent to output, and the new string (p + the last word which caused a mismatch) is added to $D(x)$. The last input word is then used as the next starting point: in this way, successively longer sequences of words are registered in the dictionary and made available for subsequent encoding, with no repeated entries in $D(x)$. An example for the encoding of the string "TO BE OR NOT TO BE OR NOT TO BE OR WHAT" after tokenization is reported in Table 1. It helps to remark that the output of the simulated compression process is not of interest for us, as the only thing that will be used is the dictionary.

Table 1
LZW encoding of the tokens composing the string "TO BE OR NOT TO BE OR NOT TO BE OR WHAT". The compressor tries to substitute pattern codes referring to sequences of words which occurred previously in the text.

Current token	Next token	Output	Added to dictionary
Null	TO		
TO	BE	*TO*	TO BE = ⟨1⟩
BE	OR	*BE*	BE OR = ⟨2⟩
OR	NOT	*OR*	OR NOT = ⟨3⟩
NOT	TO	*NOT*	NOT TO = ⟨4⟩
TO BE	OR	⟨1⟩	TO BE OR = ⟨5⟩
OR NOT	TO	⟨3⟩	OR NOT TO = ⟨6⟩
TO BE OR	WHAT	⟨5⟩	TO BE OR WHAT = ⟨7⟩
WHAT	♯	*WHAT*	

The patterns contained in the dictionary $D(x)$ are then sorted in ascending alphabetical order to enable the binary search of each pattern in time $O(logN)$, where N is the number of entries in $D(x)$. The dictionary is finally stored for future use: this procedure may be carried out offline and has to be performed only once for each data instance. Whenever a string x is checked against a database containing n dictionaries, a dictionary $D(x)$ is extracted from x as described and matched against each of the n dictionaries. The FCD between x and an object y represented by $D(y)$ is defined as:

$$FCD(x,y) = \frac{|D(x)| - \cap(D(x), D(y))}{|D(x)|} \quad (2)$$

where $|D(x)|$ and $|D(y)|$ are the sizes of the relative dictionaries, represented by the number of entries they contain, and $\cap(D(x), D(y))$ is the number of patterns which are found in both dictionaries. We have $FCD(x,y) = 0$ iff all patterns in $D(x)$ are contained also in $D(y)$, and $FCD(x,y) = 1$ if no single pattern is shared between the two objects.

The FCD allows computing a compression-based distance between two objects in a faster way with respect to NCD (up to one order of magnitude), as the dictionary for each object must be extracted only once and computing the intersection between two dictionaries $D(x)$ and $D(y)$ is faster than compressing the concatenation of x appended to y [6]. The FCD is also more accurate, as it overcomes drawbacks such as the limited size of the lookup tables, which are employed by real compressors for efficiency constraints: this allows exploiting all the patterns contained in a string. Furthermore, while the NCD is totally data-driven, the FCD enables a token-based analysis which allows preprocessing the data, by decomposing the objects into fragments which are semantically relevant for a given data type or application. This constitutes a great advantage in the case of plain texts, as the direct analysis of words contained in a document and their concatenations allows focusing on the relevant informational content. In plain English, this means that the matching of substrings in words which may have no semantic relation between them (e.g. 'butter' and 'butterfly') is prevented. Additional improvements can be made depending on the texts language. For the case of English texts, the subfix 's' can be removed from each token, while from documents in Italian it helps to remove the last vowel from each word: this avoids considering semantically different plurals and some verbal forms.

A drawback of the proposed method is that it cannot be applied effectively to very short texts. The algorithm needs to find reoccurring word sequences in order to extract dictionaries of a relevant size, which are needed in order to find patterns shared with other dictionaries. Therefore, the compression of the initial part of a string is not effective: we estimated empirically 1000 tokens or words to be a reasonable size for learning the model of a document and to be effective in its compression.

3. Experimental results

The FCD as described in the previous section can be effectively employed in tasks like authorship and plagiarism analysis. We report in this section experiments on five datasets written in English, Italian, German, Greek, and Spanish.

3.1. The Federalist papers

We consider a dataset of English texts known as Federalist Papers, a collection of 85 political articles written by Alexander Hamilton, James Madison and John Jay, published in 1787–88