Author Identification Using a Tensor Space Representation

Spyridon Plakias and Efstathios Stamatatos[1]

Abstract. Author identification is a text categorization task with applications in intelligence, criminal law, computer forensics, etc. Usually, in such cases there is shortage of training texts. In this paper, we propose the use of second order tensors for representing texts for this problem, in contrast to the traditional vector space model. Based on a generalization of the SVM algorithm that can handle tensors, we explore various methods for filling the matrix of features taking into account that similar features should be placed in the same neighborhood. To this end, we propose a frequency-based metric. Experiments on a corpus controlled for genre and topic and variable amount of training texts show that the proposed approach is more effective than traditional vector-based SVM when only limited amount of training texts is used.

1 INTRODUCTION

Author identification deals with the assignment of a text of unknown authorship to one author, given a set of candidate authors for whom text samples of undisputed authorship are available. The plethora of available electronic texts (e.g., e-mail messages, online forum messages, blogs, source code, etc.) indicates a wide variety of applications in areas such as intelligence, criminal law, computer forensics, etc. [1]

From a machine learning point-of-view, author identification can be viewed as a multi-class single-label text categorization (TC) task. Actually, several studies on TC use this problem as one more testing ground together with other tasks, such as topic identification, language identification, genre detection, etc. [6] However, there are some important characteristics of author identification that distinguish it from other TC tasks. In particular, in style-based TC the most important factor for selecting features is the frequency [4]. On the contrary, in topic-based TC the most frequent words are excluded since they carry no semantic information. Moreover, in the typical applications of author identification usually there is shortage of training texts for the candidate authors. This stands for both the amount and length of training texts. Therefore, it is crucial for authorship identification methods to be able to handle limited training texts effectively.

The vast majority of TC methods use a vector-based representation of texts. Traditionally, a bag-of-words approach provides several thousands of lexical features. Alternatively, character-based features (character *n*-grams) can be used. The latter have provided very good results in authorship identification experiments albeit the fact they increase considerably the dimensionality of the representation [5]. Especially in the case of short texts, such representation will produce very sparse data. Powerful machine learning algorithms such as support vector machines (SVM) can effectively handle such high dimensional and sparse data. However, in case we have only a few instances for training, such algorithms are less effective.

In this paper, we propose the use of tensor space representation for author identification tasks in order to cope with the problem of limited training texts. That is, instead of representing a text as a vector, we represent it as a matrix. Using a tensor of second order, the dimensionality of the text representation remains high but the classification algorithm has to learn much less parameters. As a result, it can better handle cases with very limited training instances. To this end, we use a generalization of the SVM algorithm that can handle tensors instead of vectors [3]. In contrast to the vector model, the position of each feature within the matrix is important since relevant features should be placed in the same row or column. Therefore, we examine several techniques for filling the representation matrix so that relevant features to be in the same neighbourhood. A set of experiments on a corpus controlled for genre and topic shows that when multiple short training texts are available the SVM model is the most effective. However, when only limited amount of short training texts is available, the tensor model produces better results.

2 THE TENSOR-BASED MODEL

In a vector space model, a text is considered as a vector in R^n, where n is the number of features. A second order tensor model considers a text as a matrix in $R^x \otimes R^y$, where x and y are the dimensions of the matrix. A vector $\mathbf{x} \in R^n$ can be transformed to a second order vector $\mathbf{X} \in R^x \otimes R^y$ provided $n \approx x*y$. A linear classifier in R^n (e.g., SVM) can be represented as $\mathbf{a}^T\mathbf{x}+b$, that is, there are n+1 parameters to be learnt (b, a_i, i=1,…,n). Similarly, a linear classifier in $R^x \otimes R^y$ can be represented as $\mathbf{u}^T\mathbf{X}\mathbf{v}+b$, that is, there are x+y+1 parameters to be learnt (b, u_i, i=1,…y, v_j, j=1,…x). Consequently, the number of parameters is minimized when x=y and this is much lower than n. Therefore, the vector space representation is more suitable in cases with limited training sets.

To be able to handle tensors instead of vectors, we use a generalization of SVM, called support tensor machines (STM) [3]. This algorithm works iteratively. First, it sets $\mathbf{u}=(1,\ldots,1)^T$. Then, it solves a standard SVM optimization problem to compute an estimation of $\mathbf{v}$. Once $\mathbf{v}$ is estimated, it solves another standard SVM optimization problem to estimate a new $\mathbf{u}$. The procedure of calculating new values for $\mathbf{u}$ and $\mathbf{v}$ is repeated until they tend to converge.

It is obvious that the tensor-based model takes into account associations between the features. Each feature is strongly associated with features that are in the same row and column. It is, therefore, crucial to place relevant features in the same neighbourhood. In conclusion, to transform suitably a vector representation to a second order tensor representation, one has to define what features are considered relevant and how relevant features are placed in the same neighbourhood.

[1] Dept. of Information and Communication Systems Eng., University of the Aegean, 83200 – Karlovassi, Greece, email: stamatatos@aegean.gr

In this paper, we consider the frequency of occurrence as the factor that determines relevance among features [4]. In a binary classification case, where we want to discriminate author A from author B, the relevance $r(x_i)$ of a feature x_i is:

$$r(x_i) = \frac{f_A(x_i) - f_B(x_i)}{f_A(x_i) + f_B(x_i) + b}$$

where $f_A(x_i)$ and $f_B(x_i)$ are the relative frequencies of occurrence of feature x_i in the texts of author A and B, respectively, and b a smoothing factor. The higher the $r(x_i)$, the more important the feature x_i for author A. Similarly, the lower the $r(x_i)$, the more important the feature x_i for author B.

In order to fill the matrix with the features taking into account the just defined relevance of features, we examined three techniques (an example for each case is shown in figure 1):

Vertical: the columns of the matrix are filled with decreasing relevance values. Hence, the first columns of the tensor will be strongly associated with author A and the last columns with author B. On the other hand, the rows of the matrix contain features of mixed importance for the two authors.

Diagonal: we start from the upper left corner of the matrix and fill diagonals with decreasing relevance values. Hence, the upper left part of the matrix will be strongly associated with author A and the lower left part with author B. That way, the first rows and columns are mainly associated with author A while the last rows and columns with author B.

Hilbert: we use the Hilbert space filling curve [2]. Examples of such curves are shown in figure 2. This technique produces small neighbourhoods of relevant features but any row or column contain features of mixed importance.

$$\underset{\textit{Vertical}}{\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\\8\\9\end{bmatrix} \rightarrow \begin{bmatrix}1&4&7\\2&5&8\\3&6&9\end{bmatrix}}, \underset{\textit{Diagonal}}{\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\\8\\9\end{bmatrix} \rightarrow \begin{bmatrix}1&3&6\\2&5&8\\4&7&9\end{bmatrix}}, \underset{\textit{Hilbert}}{\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\\8\\9\end{bmatrix} \rightarrow \begin{bmatrix}4&3&2\\5&6&1\\8&7&9\end{bmatrix}}$$

Figure 1. Three different techniques to transform a vector to a second order tensor. The vector features are sorted with decreasing relevance *r*.

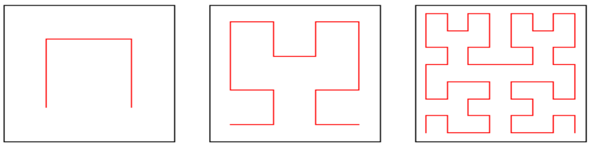

Figure 2. Examples of the Hilbert space filling curve.

4 EXPERIMENTS

The corpora used for evaluation in this study consist of newswire stories in English taken from the publicly available Reuters Corpus Volume 1 (RCV1). The top 10 authors with respect to the amount of texts belonging to the topic class CCAT (about corporate and industrial news) were selected. Therefore, this corpus of short texts is controlled for genre and topic hoping that the main factor that distinguishes the texts will be the authorship. Three versions of this corpus were formed using 50, 10 or 5 training texts per author, respectively. In all cases, the test corpus comprises 50 texts per author not overlapping with the training texts.

To represent the texts we used a character *n*-gram approach. Thus, the feature set consists of the 2,500 most frequent 3-grams of the training corpus. A standard SVM model was built using the vector of 2,500 features. Moreover, the tensor model was based on a 50x50 matrix. For each space filling technique (vertical, diagonal, and Hilbert) we built a STM model. Note that since we deal with a multi-class author identification task, we followed a one vs. one approach, that is, for each pair of authors a STM model was built and the space filling technique was based on the feature relevance for that pair of authors. Based on preliminary experiments, we set the *C* parameter of SVM to 1, the corresponding parameter for STM models to 0.1 and the smoothing parameter *b* equal to 1. The comparison of the performance of SVM and STM models can be seen in table 1. Although SVM is superior when multiple training texts are available, the STM model based on vertical space filling provides better results when the training corpus is limited.

Table 1. Performance of SVM and STM models.

Method	Training texts per author		
	50	10	5
SVM	**80.8%**	64.4%	48.2%
STM-Vertical	78.0%	**68.0%**	**51.2%**
STM-Diagonal	75.6%	60.8%	47.6%
STM-Hilbert	76.6%	66.6%	46.0%

5 CONCLUSION

In this paper, we presented a tensor-based model for the author identification problem. The proposed approach is more effective than SVM when only limited amount of training texts is available. We used the frequency as the criterion of feature relevance and examined several space filling techniques to form the feature matrix so that relevant features to be in the same neighbourhood. The vertical method seems to provide the best results for limited training corpora. This technique produces some subsets of features (columns of matrix) that are strongly associated with the authors as well as other subsets (rows) that contain features of mixed importance for the authors. Further experiments should be conducted to verify this promising result. Moreover, more complex space filling techniques can be tested to provide even better results.

REFERENCES

[1] A. Abbasi and H. Chen, ‘Applying Authorship Analysis to Extremist-Group Web Forum Messages’, *IEEE Intelligent Systems*, **20**(5), 67-75, (2005).

[2] A.R. Butz, ‘Alternative Algorithm for Hilbert’s Space Filling Curve’, *IEEE Trans. On Computers*, 20, 424-42 (1971).

[3] D. Cai, X. He, J.R. Wen, J. Han, W.Y. Ma. *Support Tensor Machines for Text Categorization*, Technical report, UIUCDCS-R-2006-2714, University of Illinois at Urbana-Champaign, (2006).

[4] M. Koppel, N. Akiva, and I. Dagan, I. ‘Feature Instability as a Criterion for Selecting Potential Style Markers’, *Journal of the American Society for Information Science and Technology*, **57**(11), 1519–1525, (2006).

[5] E. Stamatatos, ‘Ensemble-based Author Identification Using Character n-grams’, *Proc. of the 3rd International Workshop on Text-based Information Retrieval*, 41-46 (2006).

[6] D. Zhang and W.S. Lee,. ‘Extracting Key-substring-group Features for Text Classification’, *Proc. of the 12th Annual SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 474-483 (2006).