Text Document Classification and Pattern Recognition

Qin Wu, Eddie Fuller, Cun-Quan Zhang
Department of Mathematics,
West Virginia University,
Morgantown, WV 26506-6310
USA
Emails: qinwu@~~, ed@~~, cqzhang@math.wvu.edu

Abstract
In this extended abstract, a novel approach is proposed for text pattern recognition. Instead of the traditional models which are mainly based on the frequency of keywords for text document classification, we introduce a new graph theory model which is constructed based on both information about frequency and position of keywords. We applied this new idea to the detection of fraudulent emails written by the same person, and plagiarized publications. The results on these case studies show that this new method performs much better than traditional methods.

I. Introduction
Among the huge number of Internet documents, chatting log files, and archives of other digitized documents, how can we classify them into smaller categories in terms of their subject? Keywords frequency has long been used as a tool for estimating the probabilistic distribution of features in a document. A number of applications have been developed including language modeling [1], feature selection [2,3] and term weighting [4,5]. Based on the term frequency information, documents can be classified by several clustering methods such as decision trees [6], neural networks [7,8], Bayesian methods [9,10], or support vector machines [11,12,13].

Although keyword frequency is one of the most popular approaches for such classification processing, extensive studies and experiments have shown that such classification is not detailed enough to support decision makes due to the fact that the similarity between documents based on keywords frequency is too rough. This popularly used traditional method is evidently very effective for separating documents from very different categories, such as, physics, biology, social science, etc. However, it is not able to further cluster them into smaller groups. In order to provide an effective tool for decision support, this traditional method should be improved so that it can be further applied for intelligence analysis, cyber crime detection, internet monitoring, chatting log surveillance, detection of plagiarism, and many data mining problems related to security, ethical issues ([1] – [13]).

In this extended abstract, we will introduce a novel approach. By using some graph theory tools, we are able to further classify documents from the same category into smaller groups based on their writing patterns. A weighted directed graph is created for each document, it records the information not only the keyword frequency, but also their location in the document. The cosine similarity between the adjacency matrices of graphs for two documents is considered as the measurement of similarity for further clustering.

A set of algorithms for the estimation of signature vectors and clustering will be briefly presented in this paper. This algorithm has been implemented and applied to two sets of sample documents: Nigerian Fraud Emails, each of which has the same topic: to transfer money into some bank accounts in order to receive lager sum of payback [14]; Papers in academic journals of mathematics, some of which are well-known cases as plagiarized publications [15,16].

Each group is in the SAME category. The testing results show that the new method is able to detect patterns of documents written by the same author, and therefore, clearly separate samples into meaningful and accurate subgroups. However, the testing result of the traditional method shows a significantly large amount of false positive outputs.

In next section, we describe the schema for representing a document as a weighted directed multigraph. Computational complexity of the algorithm will also be presented (with no mathematical detail in this extended abstract). In section IV, we will present the outputs of this new method and the comparison with the traditional method.
II. Graph Model for Pattern Recognition

The overall approach of this algorithm begins with the identification of a set of relevant keywords. Once these are selected, we then aggregate the relative distances of the keywords with a document. This in turn is used to construct a weighted directed multigraph that generates representing vectors for each document in a high dimensional feature space. These vectors can then be used to determine similarity values for any pair of documents.

II-1. Summary of our Method

Step 1: Using weighted directed multigraph to find a signature vector for each document.

Step 2: Calculate the similarities between any two documents via their signature vectors.


Details of each step will be described step by step with a simple example.

II-2 Details of the Step 1.

To have a clear view of the algorithm, we will use the following example (see Figure 1, a sample from an open data-set source [14]) to explain the procedure.

(i) Record the keyword information appeared in the document.

For a given document, the following steps are applied to it. Suppose we have already chosen a set of words as keywords, say, \( K=\{K_1, K_2, \ldots, K_m\} \). Record every keyword and its position in the document, and the frequency of each keyword as well. For this example (Figure 1), we use the keyword set: \{bank, fund, account, transfer\}. Thus we have the tables (Table 1 and Table 2).

(ii) Construct a weighted directed multigraph

A weighted directed multigraph \( G_m \) with the vertex set \( K=\{K_1, K_2, \ldots, K_m\} \) is constructed as follows. For each
pair of keywords \( K_i, K_j \) if \( K_i \) appears at the position \( p_i \) and \( K_j \) appears at the position \( p_j \) with \( p_j > p_i \), then add an arc from the vertex \( K_i \) to the vertex \( K_j \) with the weight \( p_j - p_i \). (Note that the resulted weighted directed multigraph may contain not only parallel arcs but also loops.) For the given sample document (Figure 1), a weighted directed multigraph \( G_m \) is constructed (see Figure 2).

![Figure 2. A weighted multi-graph \( G_m \) representing the document in Fig. 1](image_url)

(iii) Simplification of representing graphs

The weighted directed multigraph \( G_m \) constructed in the previous step is simplified as follows: a directed graph \( G_s \) is constructed from \( G_m \), in which, parallel arcs are combined (see Figure 3).

![Figure 3. A (simplified) weighted graph \( G_s \)](image_url)

II-2. Details of the Step 2

(i) Signature Vector

An input document is represented by a 20-dimentional vector, in which the first 4 components are the frequencies of keywords, while the last 16 components is the \((4 \times 4)\)-adjacency matrix of \( G_s \). For the set of 500 input documents, a \((500 \times 20)\)-matrix is created where each row represents a document. The matrix is further normalized for every column of the matrix as a pre-processing. Each row \( v_i \) of the resulting matrix is called the signature vector of the corresponding document \( D_i \).

(ii) Similarity

The similarity \( s_{ij} \) between any two documents \( D_i \) and \( D_j \) is determined by the cosine similarity of the signature vectors. That is, \( s_{ij} = \cos \alpha \) where \( \alpha \) is the angle between signature vectors.

III. Computational complexity

In this extended abstract, we omit the mathematical detail of the algorithm and complexity analysis. The worst-case complexity is no more than \( O(n^2 m^2) \) where \( m \) the size of selected keywords set, \( n \) is the number input documents for comparison.

IV. Testing Results

In order to evaluate the effectiveness of our algorithm, we compare the results of our method with other method. We calculate the similarity between every pair of documents by two different ways. \( KF \): only use keyword frequency information; \( KFP \): use keyword frequency and pattern information, which is based on the weighted directed multigraph model described in this paper.

IV-1. Fraudulent emails

The traditional method (\( KF \)) does produce large amount of false positives. For example, the emails displayed in Figures 4 and 5 are obviously written in very different styles. The similarities estimated by \( KF \) and \( KFP \) methods are 1 and 0.43177, respectively.

IV-2. Plagiarism Papers

Date description: a well-known plagiarism paper [16] (named Paper-1A) on independence number of a graph and its corresponding original paper (named Paper-1B). In order to have a comparison, a set of another 35 papers from the internet (named Paper-2, Paper-3, ... , Paper-36) was downloaded randomly, which are all related to the
same subject: independence numbers of graphs. The following pictures are the first pair of pages: Paper-1A and Paper-1B (Figure 6 and Figure 7).

Figure 4. email 2002-02-20a.html

Figure 5. email 2002-07-04b.html

Short Communication

Some results on the independence number of a graph

Abstract

In this paper, we give new lower bounds for the independence number \(\alpha(G)\) of a finite and simple graph \(G\).

Keywords: Graphs, Independence number, lower bounds.

Graphs, considered here, are finite and simple (without loops or multiple edges), and [1],[2] are followed for terminology and notation. Let \(G=(V,E)\) be an undirected graph, with the set of vertices \(V=\{v_1, v_2, \ldots, v_n\}\) and the set of edges \(E\), such that \(\alpha(G)\) is the independence number of \(G\).

We denote by \(\delta(v)\) the degree of a vertex \(v\) in \(G\). It is well known (e.g., see [1]-[2]) that \(\alpha(G) \leq \delta(v)\) for any vertex \(v\) in \(G\).

Let \(\alpha(G)\) be the number of vertices having the distance \(d\) from a vertex \(v\) in \(G\) and let \(\delta(G)\) be the independence number of \(G\).

**Lemma 1.** If \(G\) is a triangle-free graph, then
\[
\alpha(G) \leq \delta(G) - \sum_{d \geq 2} \alpha(d(G)) \cdot \delta(v) - \delta(G).
\]

**Proof.** We randomly label the vertices of \(G\) with a permutation of the integers from 1 to \(n\). Let \(v \in V\) be the set of vertices \(v\) for which the minimum label on vertices at distance 1, 2 or 3 from \(v\) is on a vertex at distance 1. Obviously, the probability that \(v\) is a vertex in \(G\) is given by \(\delta(v) - \delta(v) \cdot \delta(G)\) and therefore, the expected size of \(S\) is equal to \(\delta(G)\).

Moreover, if \(S\) must be an independent set of \(G\), since, otherwise, if \(S\) is an edge of \(G\) it is easy to see that it must lie in a triangle of \(G\), contradicting the hypothesis. Thus, the lemma is proved.

**Theorem 1.** If \(G\) is a triangle-free and pentagon-free graph with \(m\) edges, then \(\alpha(G) \geq \delta(G)\).

**Proof.** Let \(d(G)\) be the average degree of vertices in \(G\). Since \(G\) is a triangle- and pentagon-free graph, then we have \(d(G) \geq \delta(G)\). By considering the neighbors of \(v\) and \(d(G) \leq \delta(v)\), for any vertex \(v\) in \(G\) and the vertices at distance \(d\) from \(v\), we get \(\alpha(G) \geq \delta(G)\).

Thus, by the above bounds, \(\alpha(G) \geq \delta(G)\) and, therefore, \(\alpha(G) \geq \alpha(G)\), the theorem being proved.

**Lemma 2.** If \(G\) is a graph with an odd girth \(2k+1\) or \(2k+2\) or greater, then
\[
\alpha(G) \geq \delta(G) - \sum_{d \geq 2} \alpha(d(G)) \cdot \delta(v) - \delta(G).
\]

**Figure 6. Paper-1A**
A set of keywords was selected which consists of 23 frequently used standard terminologies in popular graph theory textbooks [18,19]. Tables 3 and 4 indicate the significant difference in the applications of both methods: KF and KFP.

<table>
<thead>
<tr>
<th>Paper 1A</th>
<th>Paper 1B</th>
<th>Similarity by KFP Method</th>
<th>Similarity by KF Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper 1A</td>
<td>Paper 1B</td>
<td>0.77857</td>
<td>0.97074</td>
</tr>
<tr>
<td>Paper 11</td>
<td>Paper 13</td>
<td>0.34563</td>
<td>0.99500</td>
</tr>
<tr>
<td>Paper 13</td>
<td>Paper 28</td>
<td>0.20377</td>
<td>0.98567</td>
</tr>
<tr>
<td>Paper 11</td>
<td>Paper 28</td>
<td>0.09859</td>
<td>0.98011</td>
</tr>
<tr>
<td>Paper 31</td>
<td>Paper 6</td>
<td>0.05503</td>
<td>0.97190</td>
</tr>
</tbody>
</table>

Table 3. Some false positive of KF method

<table>
<thead>
<tr>
<th>Similarity between Paper</th>
<th>Similarity between all other pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>KFP method</td>
<td>all &lt; 0.6, and average &lt; 0.2</td>
</tr>
<tr>
<td>Paper 1A and Paper 1B</td>
<td>6 pairs &gt; 0.97</td>
</tr>
</tbody>
</table>

Table 4. Similarity for all sample pairs

From Table 4, estimated by KFP method, the similarity between the Paper-1A (the plagiarism paper) and Paper-1B (the original paper) is 0.78, which is the highest among all pairs, and the similarities between all other pairs are less than 0.6, most of them are far less than 0.2. However, when KF method is applied, the similarity between the 1A and 1B is 0.97 which is not the highest, and there are other 6 pairs of papers have similarities greater than 0.97 (some of those pairs are listed in Table 3).

Acknowledgements.
The authors of this paper are partially supported by a WV EPSCoR grant.

References


8) H. Ng, W. Goh, and K. Low, Feature selection, perceptron learning, and a usability case study for text categorization, in Proc. 20th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR97), pages 67-73, 1997.


10) Y. Yang and X. Liu, A re-examination of text categorisation methods, in Proc. 22nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR99), pages 67-73, 1999.


16) Wikipedia, the free encyclopedia: 
http://en.wikipedia.org/wiki/D%C4%83nu%C5%A3₂\_Marcu

