EIGENVECTOR SPACE MODEL TO CAPTURE FEATURES OF DOCUMENTS

Choi DONGJIN
Department of Computer Engineering Chosun University
Gwangju, South Korea
e-mail: Dongjin.Choi84@gmail.com
Kim PANKOO
Department of Computer Engineering Chosun University
Gwangju, South Korea
e-mail: pkkim@chosun.ac.kr

Abstract

Eigenvectors are a special set of vectors associated with a linear system of equations. Because of the special property of eigenvector, it has been used a lot for computer vision area. When the eigenvector is applied to information retrieval field, it is possible to obtain properties of documents data corpus. To capture properties of given documents, this paper conducted simple experiments to prove the eigenvector is also possible to use in document analysis. For the experiment, we use short abstract document of Wikipedia provided by DBpedia as a document corpus. To build an original square matrix, the most popular method named tf-idf measurement will be used. After calculating the eigenvectors of original matrix, each vector will be plotted into 3D graph to find what the eigenvector means in document processing.

Key-words: *eigenvector, Vector Space Model, Natural Language Processing, document analyzing, Information Retrieval, text mining*

JEL Classification: C_{83}

1. Introduction

To computer understand meaning of documents written by human, many researchers have been given great efforts to make computer think like human using lots of different methods. For example, Support Vector Machine (SVM) is the most common method to identify a feature of objects in computer vision, biomedical, and natural language processing field [1]. This method is based on the fact that every object can be considered as a vector after applying feature detection. This vector can be classified or clustered using similarity method to determine what vector meaning is. This is powerful method for long period. However, it has an unavoidable drawback that the dimension of the vector is too huge. This is the main reason why computing time takes a lot using SVM method. Therefore, it is not suitable for real time system. To overcome limitation of SVM method, LSA had been proposed [2]. LSA uses the singular value decomposition (SVD) to capture latent semantic associations which do not appeared in SVM. LSA became famous for its ability to effectively handle and capture hidden meaning of vector after splitting original vector to 3 types of vectors. This method has been popular in information retrieval field due to the fact that it is possible to extract semantic meaning of documents. However, it still has a limitation that computation and storage of LSI matrix is costly. Consider that LSA matrix (term-by-document) represents the web document. If the size of the document is big, the size of the matrix will be huge, too. In order to reduce the computation time and storage supply, this paper suggests a method to capture the feature of documents using eigenvector method. The eigenvectors of a square matrix are the non-zero vectors which, being multiplied by the matrix, remain proportional to the original vector. The eigenvectors are also called proper vectors, or characteristic vector. In other word, the eigenvector represents a property of the original vector. For this reason, this paper consists of the method to capture properties of documents based on eigenvector matrix (term-by-document) from original matrix, which includes *tf* (term frequency)-*idf* (inverse document frequency) values using Wikipedia[1] short abstract document corpus. Wikipedia is one of the most famous collaborative encyclopedia webpages. It consists of over 3.5 million documents in English and constantly managed by experts. This is the reason why this paper uses Wikipedia documents as a fundamental data set. To simplify the experiment, we focused on "computer" domain, which is a document set that includes "computer" term. There are many noises in natural documents, such as "and", "the", "an", "to" and special characters. These words have no meaning but are required for syntactic grammar rules. After filtering these noise terms, *tf-idf* matrix will be constructed in term-by-document form. According to the linear algebra theory, the eigenvector and eigenvalue matrix of original matrix will be calculated. This is the core point of this paper that the property of original matrix will be emerged in eigenvector matrix. After building eigenvector matrix, each of the vectors will be plotted into 3-D dimension space to compare feature of documents. We believe that the eigenvector space model can provide valuable features to determine which words are helpful to understand documents.

This paper is organized as follows: Section 2 describes the related works of this paper. It covers the face recognition technique using eigenvector, the eigenvector method for web information retrieval, and the method to extracting context information using eigenvector. In section 3 and 4, the proposed method of this paper will be described with examples. The evaluation and result will be presented in section 5. Finally, we conclude this paper in section 6.

2. **Related works**

The eigenvector space model has been applied for face recognition field. A collection of face images can be approximately reconstructed by storing a small collection of weights for each face [3]. Eigenfaces are sets of eigenvector used in

[1] http://www.wikipedia.org.

the computer vision problem of human face recognition. It is considered the first successful example of facial recognition technology. These eigenvectors are derived from the covariance matrix of the probability distribution of the high-dimensional vector space. Therefore, the eigenfaces provide a means of applying data compression to faces for identification purposes [3]. Also, the eigenvectors can be thought of as a set of features which together characterize the variation between face images. Because of this reason, it is possible to capture properties of documents using eigenvector. Also, this eigenvector have been applied in Information Retrieval field. Every document in World Wide Web is connected with links. These links are the important evidences to determine which documents are more related with query. PageRank is a link analysis algorithm used by the Google internet search engine. It assigns a numerical weighting to each element of a hyperlinked set of documents with the purpose of measuring its relative importance within the document set. Hundreds of thousands of links are mixed and connected without any type of routines. To capture the tendencies of links in Webpages, the eigenvector opened new research door for numerical analysts [4]. [4] focused on Web information retrieval methods such as HITS, PageRank, and SALSA using the eigenvector. The computation of PageRank is a costly, time-consuming effort that involved finding the stationary vector of a PageRank matrix. Therefore, the eigenvector can be a solution to solve time-consuming problem. The dimension of original vector will be decreased and the properties of vector will be captured because of the power of eigenvector. Also, the eigenvector can be used to extract context of document to tagging what document it is. Lee proposed method to determine keywords of a document using the eigenvector [5]. The word that has the highest eigenvalue will be selected as a keyword in Korean newspaper data corpus. As we can simply understand, the eigenvector has a great possibility to obtain property of document data sets. The eigenvector is popular in computer vision area but not in information retrieval field. Documents are also mapped into matrix the same as the matrix of a photo. In order to capture properties of documents, this paper provides, a simple method to apply eigenvector.

3. **Wikipedia Document Set**

Wikipedia is one of the most famous collaborative encyclopedia webpages. Wikipedia includes more than 3.5 million documents and is constantly expanded and adjusted by experts. Each document contained in Wikipedia consists of various forms such as title, abstract, contents information, information box, figures, category information, and core explanation. The abstract part in the Wikipedia document gives brief explanation for the title. Even though the size of abstract document is short, it contains core fact of the document. So it is proper example of fundamental data to conduct Natural Language Processing. The total size of short abstract provided by DBpedia[2] is approximately 1.3GB with 3,261,096 documents. To simplify the experiment, we focused on the computer domain, which contains

[2] http://dbpedia.org.

“computer” term in the document. The total number of the specified document set is 25,834 which is 0.008% of the original short abstract document data set. The 429 stopwords provided by Onix[3] are applied to remove unnecessary terms, which have no specific meaning. Finally, we obtained 750,591 terms after preparing test data set. Table 1 gives examples of short abstract in computer domain.

Table 1

Examples of short abstract of Wikipedia document

Short abstract of Wikipedia document
microsoft basic was the foundation product of the microsoft company. it first appeared in 1975 …
technophobia is the fear or dislike of advanced technology or complex devices, especially computers. …
in computer programming, the core language is the definition of a programming language plus …
clist (command list) (pronounced \"c-list\") is a procedural programming language for mvs/tso …
ici is a general purpose interpreted, computer programming language originally developed by …
After removing stopwords
microsoft basic foundation product microsoft company appeared altair basic basic indeed …
technophobia fear dislike advanced technology complex devices especially computers term …
computer programming core language definition programming language plus standard …
clist command list pronounced clist procedural programming language mvstso systems basic …
computer programming languages typeparameter frequentlyused generic label templates …

4. Eigenvector and Eigenvalue for Documents

Eigenvectors are a special set of vectors associated with a linear system of equations that are sometimes also known as characteristic vectors, proper vectors, or latent vectors. Each eigenvector is paired with a corresponding called eigenvalue. Mathematically, two different kinds of eigenvectors need to be distinguished: left eigenvectors and right eigenvectors. However, for many problems in physics and engineering, it is sufficient to consider only right eigenvectors. The mathematical expression of this eigenvector is as follows:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

where A is a square matrix, a non-zero vector v is an eigenvector of A if there is a scalar λ(lambda). To obtain the eigenvector of the document, documents have to be converted into matrix with term weight. The *tf-idf* (term frequency-inverse

[3] http://www.lextek.com/onix/.

document frequency) weight is a weight normally used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a corpus. The *term count* in the given document is simply the number of times a given term appears in that document. The *term frequency* is defined as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of occurrences of the considered term t_i in document d_j, and sum the number of occurrences of all terms in document d_j. The *inverse document frequency* is a measure of the general importance of the term that was obtained by dividing the total number of documents by the number of documents containing the term:

$$idf_i = \log \frac{|D|}{|\{d: t_i \in d\}| + 1}$$

where |D| is the total number of documents in the corpus and $|\{d: t_i \in d\}|$ is the number of documents where the term t_i appears. To avoid a division-by-zero, a 1 is added to the denominator. In order to obtain eigenvector of matrix A, the matrix A has to be square. To satisfy this condition, matrix A will be constructed with 10% of terms, which have the highest *df* value. Table 2 shows the example of *tf-idf* matrix.

Table 2

The example of *tf-idf* matrix

	computer	game	software	system	computers	science	university	developed	systems	games
1	0	0	0.049	0	0	0	0	0.0551	0	0
2	0	0	0	0	0.0605	0	0	0	0	0
3	0.0008	0	0	0	0	0	0	0	0	0
4	0.0007	0	0	0	0	0	0	0	0.06	0
5	0.0009	0	0	0	0	0	0	0.0699	0	0
6	0.0007	0	0	0	0	0	0	0	0	0
7	0.001	0	0	0	0	0	0	0	0	0
8	0.0008	0.0493	0	0	0	0	0	0	0	0.0789
9	0.0008	0	0.1643	0.0448	0	0	0	0	0	0
10	0.0034	0	0	0	0	0	0	0	0	0

After building *tf-idf* matrix, it is possible to obtain eigenvectors and eigenvalues corresponding to given matrix A. Table 3 and 4 gives the examples of eigenvectors and eigenvalues of *tf-idf* matrix.

Table 3

The example of eigenvalue matrix

	computer	**game**	**software**	**system**	**computers**	**science**	**university**	**developed**	**systems**	**games**
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	-0.0519	0	0	0	0	0	0	0
4	0	0	0	-0.0303	0	0	0	0	0	0
5	0	0	0	0	-0.0303	0	0	0	0	0
6	0	0	0	0	0	-0.0065	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0.0058	0	0
9	0	0	0	0	0	0	0	0	0.0612	0
10	0	0	0	0	0	0	0	0	0	0.0519

Table 4

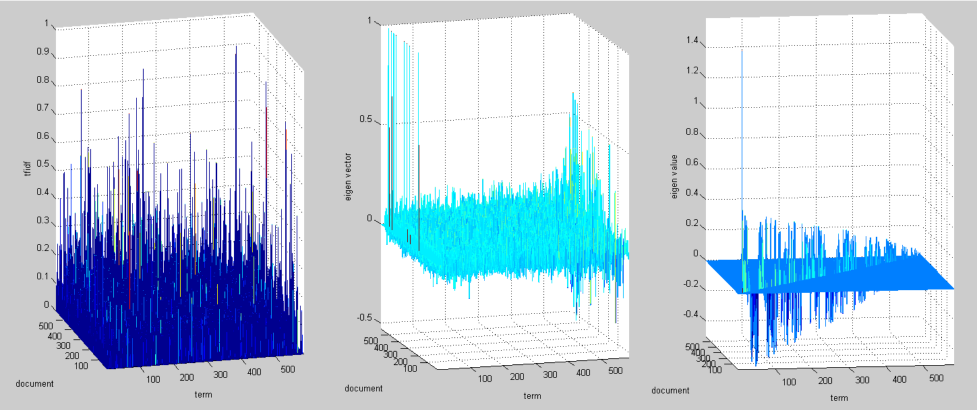
The example of eigenvector matrix

	computer	**game**	**software**	**system**	**computers**	**science**	**university**	**developed**	**systems**	**games**
1	0	0	0	-0.2233	-0.2233	0.6692	0	0.6143	-0.4274	0
2	0	0	0	0.5441	0.5441	0.5518	-0.5467	-0.5901	-0.5356	0
3	0	0	0	0.0059	0.0059	-0.0854	0	0.0875	-0.0058	0
4	0	0	0.7565	-0.0059	-0.0059	0.3067	0	-0.3369	-0.0922	-0.7565
5	0	0	0	-0.2721	-0.2721	-0.0594	0	-0.0569	-0.5418	0
6	1	0	0	0.0054	0.0054	-0.0783	-0.6122	0.0802	-0.0053	0
7	0	1	0	0.0072	0.0072	-0.1043	-0.4579	0.1070	-0.0071	0
8	0	0	0	-0.2315	-0.2315	-0.0032	0	-0.0128	-0.4688	0
9	0	0	-0.6540	0.0025	0.0025	-0.0416	0	-0.0403	-0.0888	-0.6540
10	0	0	0	00.0243	00.0243	-0.3521	0.3417	0.3610	-0.0239	0

When the eigenvector and eigenvalue are calculated by program, an imaginary number can be generated due to the mathematical procedure. The imaginary number will be removed and only the actual number will be considered because the imaginary number cannot be plotted into 3D point. The terms which have the highest absolute value of eigenvector will be selected as a context word. As table 4 shows the simple example of eigenvector, the context words of document 1 are “science”, “developed”, and “systems” even though “science” and “systems” do not occurre in document 1. If the dimension of *tf-idf* matrix is increased, the system will predict diverse context words even though those words do not occurred in the document.

5. **Experimental evaluation**

According to the previous section, the eigenvectors and eigenvalues of *tf-idf* matrix were calculated. Based on the suggested method, the proposed system will select what context words it is when the absolute value of eigenvector is in the highest range. This is the fundamental research to apply the eigenvector space model to information retrieval field to capture properties of given documents. In order to capture the properties of documents, *tf-idf* matrix, eigenvector matrix and eigenvalue matrix will be plotted into 3D graph using Matlab program. The following figure 1 indicates graphs of each matrix.



tf-idf matrix graph eigenvector matrix graph eigenvalue matrix graph

Fig. 1. *Graphs of tf-idf, eigenvector and eigenvalue matrix*

The first graph of fig. 1 indicates the *tf-idf* values of documents when *df* is bigger than 200. Because of the limitation of memory resource size, it is not able to construct the entire *tf-idf* matrix of filtered Wikipedia short abstract documents. To simplify the experiment, every term that has *df* value is less than 200 in given documents will not be considered as a candidate term. Only 583 terms are remained after applying threshold value as 200. It is easy to see that the *tf-idf* graph was is too much complicated to capture the properties of given documents even though the graph based on only 583 kinds of terms. However, the second graph of fig. 1 is different. The graph of eigenvector matrix involves particular patterns that are similar with signal data of voice. The graph goes up suddenly around the 500th term which means that those terms are closely related with “computer” query. We believe that it is possible to obtain properties of given documents through analyzing this inverted eigenvector frequency. Also, the context words for documents will be determined based on the suggested method.

6. **Conclusions and future works**

This paper contained a method to capture properties of Wikipedia short abstract documents through finding eigenvector of matrix with *tf-idf* values. The eigenvector have been applied in computer vision area not much in information retrieval field. To apply the eigenvector model to information retrieval, this paper conducted simple experiments to prove that the eigenvector is also possible to use in document analysis. The candidate terms for context words are selected from documents if their *df* values are higher than 200 due to the limitation of memory size. After comparing each of *tf-idf* matrix graph, eigenvector matrix graph, and eigenvalue matrix graph, it has been proved that the eigenvector has more evidences to capture properties of given documents. Also, it is possible to obtain context words which are the representative terms of document using this method. Therefore, this great potential to analyze huge documents for semantic information processing or text mining using the eigenvector. However, the eigenvector will be changed if data of the original matrix has is changed. This means that the eigenvector depends on the weight measuring method. This paper is only using most popular method named *tf-idf* measurement, yet. There are many different term weighting methods for document processing. In order to guarantee its potential and usability in near future these weighting methods using the eigenvector have to be compared.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0011656)

REFERENCES

William S. Nobel, *What is a support vector machine*?, 1565-1567 (Nature Biotechnology, 2006), 24.

Thomas K. Landauer, Peter W. Foltz, and Darrell Lahm, *An Introduction to Latent Semantic Analysis*, 259-284 (Discourse Processes, 1998), 25.

Matthew A. Turk and Alex P. Pentland, *Face recognition using eigenfaces*, 586-591 (IEEE Comput. Sco. Press, 1991), 3.

Amy N. Langville and Carl D. Meyer, *A Survey of Eigenvector Methods for Web Information Retrieval*, 135-161, (SIAM, 2005), 47.

Chang-Beom Lee, Min-Soo Kim, Ki-Ho Lee, Guee-Sang Lee, and Hyuk-Ro Park, *Document Thematic words Extraction Using Principal Component Analysis*, 747-754 (KIISE, 2002), 29.