A Clustering Method for Web Mining Based on Probabilistic Latent Semantic Indexing

Jianxiong Yang* and Junzo Watada*

Abstract: Exploring an intranet or internet database enables us to discover useful knowledge. In this process, a search engine plays a pivotal role. To this end, various search engines have been proposed to heighten information accuracy by exploiting key content relations in semantic web resources. But a general-purpose search engine always includes useless or irrelevant web pages in the search results. The next generation of web architecture, known as Semantic Web, can build a layered architecture to possibly mitigate this deficiency by decreasing the noisy data in a searched result. The objective of this paper is to propose a Probabilistic Latent Semantic Indexing (PLSI) method used in semantic web search engines. The method can better return appropriate information for user queries; in particular, a novel ranking strategy is provided to measure the relevance score of an annotated set of web results by considering user queries, data annotation, and the underlying ontology.

Key Words: latent semantic indexing, probabilistic latent semantic indexing, search engine, data mining, semantic web.

1. Introduction

Most web information is presented in natural languages. This formatting is convenient for users to read and view, but it can be difficult for computers to process. It also restricts simple keyword search engines in terms of their indexing capability, because these engines cannot infer meaning. For example, a human can refer to the meaning of word “pet” from the word “pets”. Because a human can understand the proper meaning from the context. But a computer cannot apprehend the meaning of the word “pets”. It can extract some strings which include “pets” only. So the word “pet” must be ignored.

Semantic Web offers a way to address this problem at the architecture level [1]. In fact, Semantic Web possesses the semantic metadata of each Web page [2].

In this paper, we prove that embedding target data into a data set can effectively extract the user’s target data using Semantic Web search engines. This sort of ranking exploits the evaluation of accurate information on a Web page. It can be used in conjunction with other ranking strategies to further improve the accuracy of retrieved results. Comparing with other ranking methods for Semantic Web, our approach depends on the user query, the ranked web pages and the underlying ontology. Thus, it allows us to effectively manage the search space and reduce the complexity of the ranking task.

In contrast, if we want to find some pet’s goods shop in Japan, we may submit search string “pet’s goods shop in Japan” or “pet’s goods shop Japan” to a general search engine, we can observe that each element is scattered in the ranked list of searched results, which often includes a large number of noised results. The major search engines have this problem, since they weave relevant documents among documents of different topics. However, given the limited space of a single result page, certain topics will be inevitably hidden from the user. Nevertheless, PLSI can improve this task by clustering items.

In this work, a novel ranking strategy is presented to measure a relevance score for Web data contained in an annotated search result by considering the user query, data annotation, and the underlying ontology.

This paper is organized as follows. In Section 2, we explain the vector space model and Latent Semantic Indexing (LSI). In Section 3, efficiency and implementation issues are discussed with regard to PLSI, and in Section 4, PLSI applications in accident evaluation are presented. In Section 5, experimental results and analysis are presented. Lastly, Section 6 offers some conclusions.

2. A Review of Past Studies

Based on fuzzy linguistic regression analysis, linear model approaches have been proposed to mine knowledge using human evaluation [3]. Its solution method is provided in the paper [4]. Evaluating ranking with a dual-scaling method enables us to mine knowledge from text databases [5]. Annotations are based on classes of concepts and the relations among them. The annotation is usually expressed through an ontology that provides a common understanding of terms within a given domain [6].

2.1 LSI

LSI is an indexing and retrieval method that uses a mathematical technique called Singular Value Decomposition (SVD) to identify relational patterns between terms and concepts contained in an unstructured collection of texts. LSI is based on the principle that words tend to have similar meanings in the same context. It requires relatively high computational performance and a large amount of memory, compared with that required by other information retrieval techniques [7]. A key feature of LSI is its ability to extract the conceptual content of texts by establishing associations among those terms.
that occur in similar contexts.

2.2 Application of LSI

LSI is used to perform automated document categorization. Several experiments have demonstrated a number of successful correlations between LSI and human inference, and these experiments have effectively categorized texts [8]. Document categorization is the assignment of documents to one or more predefined categories based on their similarity to the conceptual content of categories [9]. LSI uses template documents to establish a conceptual basis for each category. During the process of categorization, the concepts contained in the categorized documents are compared with those in the template items, and a category (or categories) is assigned to the documents based on similarities between the concepts that they contain and those that are contained in the template documents.

LSI can effectively realize dynamic clustering based on the conceptual content of the documents. Clustering is a way to group the documents based on their conceptual similarity to each other without using template documents to establish the conceptual basis for each cluster. This is very useful when dealing with an unknown collection of unstructured text [10].

LSI is not restricted to words. It can also process arbitrary character strings. Any object expressed in text can be represented in an LSI vector space. For example, texts with MEDLINE abstracts have shown that LSI is able to establish a conceptual basis for each cluster. This is very useful when dealing with an unknown collection of unstructured text [10].

LSI has proven to be a useful solution to a number of conceptual matching problems [11],[12]. The technique has been shown to capture key relationship information, including causal, goal-oriented, and taxonomic information [13].

3. A Vector Space Model and LSI

LSI can correlate semantic terms in a collection of texts. The method was first applied to texts at Bell Laboratories in the late 1980s, during which time it was also called Latent Semantic Analysis (LSA). This method can uncover the underlying latent semantic structure in word usage in the content of text and then extract the meaning of the text in response to user queries, which are commonly referred to as concept searches. Queries, or concept searches, depend on a set of documents that have undergone LSI, and they will return results that are similar in meaning to the search criteria, even if the results do not share specific words with the search criteria [14].

3.1 SVD

In linear algebra, the SVD is an important factorization method for a rectangular real or complex matrix that has many applications in signal processing and statistics. Assume $A$ is an $m \times n$ matrix with either real or complex numbers as entries. Then there exists a factorization of the form

$$A = U \Sigma V^T$$

where $U$ denotes an $m \times m$ unitary matrix, matrix $\Sigma$ denotes an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal, and $V^T$ denotes an $n \times n$ unitary matrix, that is, the conjugate transpose of $V$. This factorization is called the SVD of $A$. The diagonal entries $\sigma_i$ of $\Sigma$ are known as the singular values of $A$. Figure 1 shows its structure.

To obtain a simplified vector space, we merely include the non-zero singular values of $\Sigma$. The rank $r$ is the number of non-zero singular values of $\Sigma$.

A common usage is to place the singular values in descending order. In this case, the diagonal matrix $\Sigma$ is uniquely determined by $A$.

Then, we obtain the following new equation (2) as shown in Fig. 2.

$$A = U_r \Sigma_r V^T_r$$

where $U_r$ denotes an $m \times r$ submatrix of unitary matrix $U$, $V^T_r$ denotes an $r \times n$ submatrix of unitary matrix $V$, and $\Sigma_r$ consists of only the non-zero singular values of $\Sigma$. The rank $r$ is the number of non-zero singular values of $\Sigma_r$.

3.2 An Example of SVD

This example has terms 1, 2, 3, 4 and 5 as follows.

Term 1 appears 1, 0, 0, and 0 times in documents 1, 2, 3, and 4, respectively.

Term 2 appears 0, 0, 4, and 0 times in documents 1, 2, 3, and 4, respectively.

Term 3 appears 0, 3, 0, and 0 times in documents 1, 2, 3, and 4, respectively.

Term 4 appears 0, 0, 0, and 0 times in documents 1, 2, 3, and 4, respectively.

Term 5 appears 2, 0, 0, and 0 times in documents 1, 2, 3, and 4, respectively.

The factor matrix $A$, that is, term-document matrix $A$, is shown as

$$A = \begin{pmatrix}
1 & 0 & 0 & 0 & 2 \\
0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 4 & 0 & 0 & 0
\end{pmatrix} \begin{pmatrix}
d_1 \\
d_2 \\
d_3 \\
d_4
\end{pmatrix}$$

where $t_1, t_2, t_3, t_4$, and $t_5$ denote term 1, term 2, term 3, term 4, and term 5, respectively, and $d_1, d_2, d_3, d_4$ denote document 1, document 2, document 3, and document 4, respectively.
The document decomposition matrix is calculated in the following steps.

Step 1:
Calculate the following interactive matrix
\[ A \cdot A^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \]
\[ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 4 \end{pmatrix} \]

Step 2:
Calculate the eigenvalues and singular vectors of \( A \cdot A^T \). The non-zero singular eigenvectors are shown in descending order as follows:
\[ \lambda_1 = 16, \quad \lambda_2 = 9, \quad \lambda_3 = 5. \]
The non-zero singular values are as follows:
\[ \delta_1 = 4, \quad \delta_2 = 3, \quad \delta_3 = \sqrt{5}. \]
The rank \( r \) of the non-zero singular matrix \( \Sigma \) is 3. Then, the eigenvectors are as follows:
\[ v_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad v_3 = \begin{pmatrix} \sqrt{0.2} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \]
The singular value matrix \( \Sigma \) and the term decomposition matrix \( V \) can be written as follows:
\[ \Sigma = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix}, \quad V^T = \begin{pmatrix} 0 & 1 & 0 & 0 & \sqrt{0.2} \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.8} & 0 & 0 & 0 \end{pmatrix}. \]

Step 3:
Calculate the document decomposition matrix \( U_r \) as follows.
\[ u_1 = \frac{1}{\delta_1} A v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \]
\[ u_2 = \frac{1}{\delta_2} A v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \]
\[ u_3 = \frac{1}{\delta_3} A v_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \]
\[ U_r = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \]

If we want to analyze document attributes, we calculate the document decomposition matrix \( U_r \).

4. Probabilistic Latent Semantic Indexing (PLSI)

4.1 The PLSI Model

Probabilistic Latent Semantic Indexing (PLSI) is a probabilistic and statistical technique for analyzing co-occurrence and co-occurrence data. PLSI is expanded from LSI with a connotative probabilistic model. PLSI has applications in information retrieval and filtering, natural language processing, machine learning from text, and related areas. Thomas Hofmann [15] introduced PLSI in 1999 as a method related to non-negative matrix factorization [16].

Compared with standard latent semantic analysis, which stems from linear algebra and downsizes the occurrence tables, PLSI is based on mixture decomposition derived from a latent class model. Thus, its foundation is in statistics.

A generative model for the observation pair \((w, d)\) can be defined as follows:
- Select a document \(d\) with probability \(P(d)\) which is probability of being selected in all documents;
- Generate a latent class \(z\) with probability \(P(z|d)\) which is the distribution of document \(d\) for the latent class \(Z\);
- Generate a word \(w\) with probability \(P(w|z)\) which is the distribution of latent class for the word \(w\).

The distribution of word-document are obtained by a combination of the aspects \(P(w|z)\). Documents are not assigned to clusters. They are characterized by a specific mixture of factors with weights \(P(z|d)\). These mixing weights offer modeling flexibility.

Considering observations in the form of the co-occurrence \((w, d)\) of words \((W = \{w_1, w_2, w_3, \cdots, w_i\})\) and documents \((D = \{d_1, d_2, d_3, \cdots, d_j\})\), PLSI models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions as follows:
\[ P(w_j|d_i) = \sum_{k=1}^{c} P(w_j|z_k)P(z_k|d_i). \]

In this latent variable model for general co-occurrence data, an latent class variable \(z\in Z = \{z_1, z_2, z_3, \cdots, z_c\}\) is regarded as being unobservable.

This formulation is a symmetric formulation, where \(w\) and \(d\) are both generated from the latent class \(z\) in similar ways using the conditional probabilities \(P(w_j|z)\) and \(P(d_i|z)\), respectively. For each document \(d\), a latent class is chosen conditional on the document according to \(P(z|d)\), and a word is then generated from that class according to \(P(w|z)\). Figure 3 shows the relationship among words, latent classes, and documents.

These circle notations and lines represent the PLSI model. The left circles are words (or terms) drawn from the distribution of words for the latent class \(Z\): \(P(w|z)\), the middle circles are topics drawn from the distribution of the latent class for the documents \(P(z|d)\), and right circles are the document variables. \textbf{Word} and \textbf{Document} are observable variables; \textbf{Latent Class} relates a set of observed discrete multivariate variables to a set of latent variables.

4.2 Expectation Maximization Algorithm

Usually, the procedure used for maximum possibility estimation in latent variable models is the Expectation Maximization
(EM) algorithm. EM consists of two steps: Step 1: The Expectation step (E-step) computes posterior probabilities for the latent variables \( z \) with current estimates of the parameters as follows:

\[
P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{|z|} P(w_j|z)P(z|d_i)}.
\]

(6)

Step 2: The Maximization step (M-step) updates parameters for the given posterior probabilities computed in the previous step as follows.

\[
P(w_j|z_k) = \frac{\sum_{i=1}^{n} a(d_i|w_j)P(z_k|d_i, w_j)}{\sum_{j=1}^{m} \sum_{i=1}^{n} a(d_i|w_j)P(z_k|d_i, w_j)},
\]

(7)

\[
P(z_k|d_i) = \frac{\sum_{j=1}^{m} a(d_i|w_j)P(z_k|d_i, w_j)}{a(d_i)}.
\]

(8)

where \( a(d_i) \) denote the times of appearance document \( i \) appear in the latent class \( z \). When the expectation value \( L \) is lower than the threshold, the iteration is terminated, and we obtain an optimal solution as follows.

\[
E(L) = \sum_{i=1}^{n} \sum_{j=1}^{m} a(d_i, w_j) \sum_{k=1}^{K} P(z_k|d_i, w_j) \log[P(w_j|z_k)P(z_k|d_i)]
\]

(9)

where \( a(d_i, w_j) \) is the degree of word-document similarity.

Alternating (6) with (7) and (8) defines a convergent procedure that approaches a local maximum of the log-likelihood in (9).

The PLSI model can be used to replace the original term-document representation by a representation in a low-dimensional “latent” space in order to perform term clustering or document retrieval. The components of the document in the low-dimensional space are \( P(z = k|d) \), \( k = 1, 2, \ldots, K \) and for each unseen document or query the aforementioned components are computed by maximizing the log-likelihood with \( P(z = k) \) fixed. In particular, PLSI is found to perform well even in the cases where LSI fails completely.

5. A PLSI-Based Method for Accident Evaluation

5.1 Data Preprocessing

First, we remove all words that have no specific meaning related to accidents. For example, these words include pronouns, helping verbs, and articles. Because these words have no meaning for our purpose, we may ignore them as a noise. There still remain a number of significant text words, such that the dimensions of the space are considerably large. Therefore, we must implement a method for dimension compression. In this paper, we use the Document Frequency (DF) method to compress the space by selecting only those words with values above a threshold. Then we create the word-document matrix \( A = [a_{ij}] \) after data standardization [6], where \( 0 \leq a_{ij} \leq 1 \), which is the weight of the \( i \)th word of the \( j \)th document. \( i \) and \( j \) denote the quantity of words and documents, respectively. Usually, every word appears in a small number of documents. Thus, the word-document matrix \( A \) must be a high-order matrix. As such, we must transform \( a_{ij} \) to log\((a_{ij} + 1)\) and then divide it by its entropy. After transforming the information entropy, we obtain the following word-document matrix.

\[
A' = [a'_{ij}]
\]

where

\[
a'_{ij} = \frac{\log(a_{ij} + 1)}{-\sum_{w_j} \left[ \frac{a_{ik}}{\sum_{j=1}^{m} a_{jk}} \right] \times \log \left[ \frac{a_{ik}}{\sum_{j=1}^{m} a_{jk}} \right]}
\]

Finally, we use the LSI method to derive the word-document matrix \( A' \), and then we use the SVD method to evaluate a approximate matrix \( A_k' \). In this way, we remove noise from the word-document matrix and emphasize the relationships between words and documents. Moreover, the vector space is significantly reduced, and the accuracy of the clustered documents is increased.

5.2 Clustering Analysis

Using PLSI, we can obtain the conditional probability \( P(z_k|d_i) \) between implicit variables \( z_k \) and \( d_i \). Thus, we can create the document-implicit variable vector

\[
d_i = (P_{j,k-1}, P_{j,k+1}).
\]

(10)

where \( P_{j,k} \) denotes the conditional probability \( P(z_k|d_i) \) between the document-implicit variable \( z_k \) and document \( d_i \). This formulation reflects the relationship between document-implicit variables. Using this variable, we can estimate the similarity grade of two documents as follows.

\[
\text{Sim}(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i||_2 \cdot ||d_j||_2}
\]

where

\[
d_i \cdot d_j = \sum_{m=1}^{k} p_{i,m} p_{j,m},
\]

\[
||d_i||_2 = \sqrt{\sum_{m=1}^{k} p_{i,m}^2}
\]

5.3 The Analysis Stage

To create an efficient mining system, the following steps are used to implement a ranking method for web mining each document.
Input: Dimension $k$, threshold $\lambda$ and impliciative factor $K$ of singular matrix.

Output: The clustering result $DC = \{DC_1, DC_2, \ldots, DC_n\}$, where $n$ indicates the amount of clustered data and $DC_n$ indicates the $n$th accident case.

Step 1: Remove all irrelevant words from the text to create a word-document matrix $A$;

Step 2: Insert sample data for extraction into word-document matrix $A$;

Step 3: Create the word-document matrix $A$;

Step 4: Use SVD to obtain a $k$-rank approximate matrix $A_k$;

Step 5: Use PLSI to obtain the document-implicit variable vector $dl$ from the word-document matrix $A_k$;

Step 6: Evaluate all similarity values $\text{sim}(d_i, d_j)$ for $dl$;

Step 7: Create the maximal tree using the similarity $\text{sim}(d_i, d_j)$;

Step 8: Output the clustering results;

Step 9: Based on the output results, extract all groups that include sample data according to $A$. These data groups are desired data for us.

6. Experiment Result and Analysis

The performance of this method was evaluated using an accident database from the National Institute of Technology Evaluation (NITE). There are three types of attributes: economic loss, disaster area, and casualty state. These are used to evaluate the severity of the accident. We included some cases of serious accidents, such as cases with more than $200$ ml, in economic loss, a disaster area greater than $600$ m², or more than $50$ casualties. The main characteristics of the database used in our experiment are summarized in Table 1. The first column of this table provides the name of the data set, whereas the other columns indicate the numbers of cases and matching cases in reality.

Table 1 Database used in the experiment.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Cases</th>
<th>Matching cases in reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric appliance</td>
<td>200</td>
<td>23</td>
</tr>
<tr>
<td>Oil appliance</td>
<td>200</td>
<td>13</td>
</tr>
<tr>
<td>Gas appliance</td>
<td>200</td>
<td>11</td>
</tr>
</tbody>
</table>

After embedding the sample data into the NITE database, we create a 600-rank word-document matrix. Then we use SVD to obtain an approximate 600-rank matrix. To improve system accuracy, we use PLSI to improve data extraction. We obtain a document-implicit variable vector, and then we evaluate all similarity values for all accident cases that include sample data. At last, if any case can be clustered with sample data case, then this satisfies the criteria for inclusion [6]. Table 2 indicates the amount and percentage of all corresponding cases with each matching value. The calculation of accuracy rate is shown as:

$$A_k = (1 - |D_r - D_o|/D_o) \times 100\%,$$

where $D_r$ is the quantity of the experiment result, $D_o$ is the original quantity. It is not surprising that rule pruning extracts the closest cases besides the probable cases. The matching value can be used to sort information and thus easily obtain the target results.

7. Conclusions

The next generation of web structure, as represented by Semantic Web, provides an adequate instrument to improve search strategies and enhances the quality of user queries without requiring tiresome manual refinement [17]. However, the actual methods for ranking the returned results must be adjusted to fully exploit the documents characterized by PLSI, including ontology-based concepts and relations. Several ranking algorithms have been proposed for the use of Semantic Web to exploit relation-based metadata. Nevertheless, they mainly use data-relevance criteria based on information that must be derived from an entire knowledge base. In this work, the authors have proposed a novel ranking strategy that is capable of providing a relevance score for web data using an annotated result set by considering the user query, the data annotation, and the underlying ontology.

Data relevance is measured through a probabilistic approach that is based on several factors. The cost of the returning query results would be lowered if the annotated resources are not further processed beyond the method presented here. Further efforts are required to allow scalability within semantic web repositories that include multiple ontologies and are characterized by a large amount of data; indeed, this may alter next-generation semantic mining techniques.

Currently, the authors are testing PLSI techniques for mining web documents. In addition, they also plan to apply these PLSI techniques to web databases. They firmly believe that semantic-based mining is a promising approach to significantly unleashing the power of both the Internet and multimedia technology.

Table 2 Comparison of results.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>LSI Quantity (Accuracy rate)</th>
<th>PLSI Quantity (Accuracy rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric appliance</td>
<td>27(82.61%)</td>
<td>21(91.30%)</td>
</tr>
<tr>
<td>Oil appliance</td>
<td>11(84.62%)</td>
<td>12(92.31%)</td>
</tr>
<tr>
<td>Gas appliance</td>
<td>9(81.82%)</td>
<td>12(90.91%)</td>
</tr>
</tbody>
</table>

References


