J. Korean Math. Soc. 36 (1999), No. 3, pp. 609–619

LATENT SEMANTIC INDEXING AND LINEAR RELEVANCE FEEDBACK IN TEXT INFORMATION RETRIEVAL THEORY

KICHOON YANG

ABSTRACT. We give a mathematically rigorous description of the recently popular latent semantic indexing (LSI) method in text information retrieval theory. Also, a related problem of finding a document ranking function in linear relevance feedback is discussed.

1. Introduction

Text information retrieval focuses on the problem of retrieving documents relevant to a user need represented by a set of keywords known as a *query*. In the popular vector space model of information retrieval both documents and the query are represented as vectors in the *keyword vector space*; then the relation between a document and the query is measured either by their dot product or other related measures such as the cosine of the angle between the two vectors. However, an implementation of the vector space model relying solely on keyword matching often fails to find relevant documents or return too many irrelevant documents. One reason for this failure is the so called synonymy problem, meaning that a single concept or object has many different terms associated with it. For example, several empirical studies show that the likelihood of two people choosing the same keyword for a familiar object is less than 15%. The *latent semantic indexing* (*LSI*) method is an attempt to solve the synonymy problem whilst staying within the vector space model framework. Unlike many previous attempts the latent indexing method is more automated and numerically simpler. In this paper, we give a mathematically

Received December 28, 1998.
1991 Mathematics Subject Classification: 94A15.
Key words and phrases: singular value decomposition, latent semantic indexing, information retrieval.

rigorous description of the latent semantic indexing method and identify several problems.

One important task in information retrieval is to produce a ranking of the documents according to user preference so that the user will be able to find the required information by looking only at the top several documents. The problem, however, is that the user preference is never fully known to the retrieval system since the user can not fully specify his preference without having examined all the documents in the collection. This motivates a recursive strategy where the retrieval system learns about the user preference from a training set of sample documents. Such a system can learn and refine the *document ranking function* it uses to rank the documents; once a satisfactory set of documents has been retrieved the user can stop the recursion. In the paper, we discuss one such recursive strategy.

2. The latent semantic indexing method

Suppose we are given a document collection $\Delta = \{D_i,\ 1 \leq i \leq d\}$ and a set of keywords $\Theta = \{T_j,\ 1 \leq j \leq d\}$ used to describe the documents. We identify the keywords T_j with the canonical basis vectors δ_j, $1 \leq j \leq t$, of $\mathbb{R}^t$. Then a document or a query D is identified with a vector, which we again denote by D, in $\mathbb{R}^t$ as follows:

$$D = \left(D^j\right) = D^j \delta_j \in \mathbb{R}^t,\ D^j = \text{the } \textit{term weight} \text{ of } T_j \text{ in } D,$$

where the term weight measures the importance of a term in the document. There are many different ways to assign term weights. For example, one may simply define D^j to be the number of times T_j appears in the document.

Once a term weighting scheme has been chosen there arises the following $t \times d$ matrix called the *term-document matrix*:

$$X = \begin{bmatrix} D_1^1 & \cdots & D_d^1 \\ \vdots & & \vdots \\ D_1^t & \cdots & D_d^t \end{bmatrix} = (D_1, \cdots, D_d) \in M\,(t \times d)\,.$$

In the above, $M\,(t \times d)$ denotes the set of all $t \times d$ matrices.

A query vector, also called a pseudo-document, is a vector $q \in \mathbb{R}^t$ that is given by the user. Two most popular ways of comparing documents with the query vector are the cosine similarity and the dot product

similarity. In the case of cosine similarity one looks at the cosine of the angle between the query vector and each document in Δ, whereas in the dot product similarity the documents are ranked in order of the dot product $D_i \cdot q$. We will work with the dot product similarity throughout - we do this largely for the sake of expositional simplicity rather than for substantive reasons.

Let

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \quad r = rank\,(X)\,,$$

denote the nonzero singular values of X, and consider the singular value decomposition of X:

$$X = U\Sigma V^t,$$

where $U \in M\,(t \times r)$ with orthonormal columns, $V \in M\,(d \times r)$ with orthonormal columns, and $\Sigma \in M\,(r \times r)$ is the positive definite diagonal matrix given by

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}.$$

DEFINITION 1. The k-th singular approximation of X, where $k < r$, is defined to be

$$X_k = U_k \Sigma_k V_k^t,$$

where $U_k \in M\,(t \times k)$ and $V_k \in M\,(d \times k)$ are obtained simply by removing the last $(r - k)$ columns and Σ_k is the $k \times k$ diagonal matrix consisting of the first k singular values in the diagonal entries.

Let (ε_i) denote the canonical basis vectors for $\mathbb{R}^d$ and put

$$s = (s_{ij}) = (X\,(\varepsilon_i) \cdot X\,(\varepsilon_j)) = X^t X.$$

Given a linear surjection

$$\phi : \mathbb{R}^t \longrightarrow \mathbb{R}^k$$

we set

$$s_\phi = (s_{\phi ij}) = (\phi\,(X\,(\varepsilon_i))\,, \phi\,(X\,(\varepsilon_j)))\,.$$

We define the distance between s and s_ϕ to be the Euclidean distance in $\mathbb{R}^{d^2}$, i.e.,

$$d(s, s_\phi) = \sqrt{\sum_{i,j} (s_{ij} - s_{\phi ij})^2}.$$

We then have the following remarkable result regarding the k-th singular approximation.

THEOREM 2. *Let $\phi : \mathbb{R}^t \longrightarrow \mathbb{R}^k$ be given by the matrix U_k^t, where $U_k \in M(t \times k)$ is the matrix arising in the k-th singular approximation of X. Then $d(s, s_\phi)$ is minimal as ϕ varies over all linear surjections $\mathbb{R}^t \longrightarrow \mathbb{R}^k$.*

In the statement of the above theorem and in the preceding discussion we have used the following standard identifications:

$$M(m \times n) = Hom(\mathbb{R}^n, \mathbb{R}^m),$$

$$M(m \times n) = \mathbb{R}^{mn}.$$

The latter identification is made simply by ordering the matrix entries lexicographically. To make the former identification explicit, let $(\varepsilon_i)_{1 \le i \le n}$ denote the canonical basis of $\mathbb{R}^n$ and also let $(\delta_j)_{1 \le j \le m}$ denote the canonical basis of $\mathbb{R}^m$. Then given any linear map $\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ we can associate an element (ϕ_i^j) of $M(m \times n)$ as follows:

$$\phi(\varepsilon_i) = \sum_j \phi_i^j \delta_j.$$

The above theorem follows essentially from the following two propositions.

PROPOSITION 3. *Let X_k be as in the above. Then $d(X, X_k)$ is minimal as X_k varies over all $t \times d$ matrices of rank k, where d denotes the Euclidean distance in $\mathbb{R}^{td}$.*

PROPOSITION 4. *Let $\phi = U_k^t : \mathbb{R}^t \longrightarrow \mathbb{R}^k$, and consider $\hat{X} = U_k^t X \in M(k \times d)$. Then $\hat{X}^t \hat{X} = X_k^t X_k \in M(d \times d)$.*

The above theorem provides a mathematical basis for the LSI method. By taking k strictly smaller than the rank of the term-document matrix and replacing the term-document matrix accordingly, one is able to take

into account the *underlying semantic structure* instead of considering merely the raw term matches between the query and the documents.

As an example, consider the following document collection - this example appears in [Dumais et al.]. There are nine documents and twelve keywords, having to do with two rather different topics, computer-human interface and mathematical graph theory. The document collection is given by:

- c1: *Human* machine *interface* for Lab ABC *computer* applications
- c2: A *survey* of *user* opinion of *computer system response time*
- c3: The *EPS user interface* management *system*
- c4: Systems and *human systems* engineering testing of *EPS*-2
- c5: Relation of *user*-perceived *response time* to error measurement
- m1: The generation of random, binary, unordered *trees*
- m2: The intersection *graph* of paths in *trees*
- m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
- m4: *Graph minors*: A survey

The twelve keywords are:

- t1: *human*
- t2: *interface*
- t3: *computer*
- t4: *user*
- t5: *system*
- t6: *response*
- t7: *time*
- t8: *EPS*
- t9: *survey*
- t10: *tree*
- t11: *graph*
- t12: *minor*

Note that if a user requested papers dealing with "human computer interaction," a keyword-based retrieval system would return titles *c*1, *c*2, and *c*4, since these titles contain at least one keyword from the user query. However, *c*3 and *c*5 would not be returned, since they share no words with the query.

Using the raw term-frequency indexing we obtain the following $t \times d$ term-document matrix X with $t = 12$ and $d = 9$:

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Note that the query vector corresponding to "human computer interaction" is given by

$$q = (1, 0, 1, 0, \cdots, 0)^t \in \mathbb{R}^{12},$$

and that

$$q \cdot A_i = 0 \text{ unless } i = 1, 2, 4.$$

Indeed $q \cdot A_1 = 2$ and $q \cdot A_2 = q \cdot A_4 = 1$.

Calculations show that the rank of X is full and that its two largest singular values are 3.34 and 2.54.

The second singular approximation of X is given by:

$$X_2 = \begin{bmatrix} 0.22 & -0.11 \\ 0.2 & -0.07 \\ 0.24 & 0.04 \\ 0.4 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.3 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{bmatrix} \begin{bmatrix} 3.34 & 0 \\ 0 & 2.54 \end{bmatrix} \begin{bmatrix} 0.2 & -0.06 \\ 0.61 & 0.17 \\ 0.46 & -0.03 \\ 0.54 & -0.23 \\ 0.28 & 0.11 \\ 0 & 0.19 \\ 0.01 & 0.44 \\ 0.02 & 0.62 \\ 0.08 & 0.53 \end{bmatrix}^t.$$

From this we calculate the new dot matrix similarity measures for the nine documents:

$$(q \cdot X_{2i}) = \begin{pmatrix} 0.31 & 0.91 & 0.74 & 0.88 & 0.42 & -0.03 & -0.06 & -0.07 & 0.03 \end{pmatrix}.$$

Thus the relevant documents, with respect to the new term-document matrix X_2 and in the order of their relevance, are $c2$, $c4$, $c3$, $c5$, and $c1$.

Note that if k is too large (i.e., too close to r), then we lose the latent semantic structure; on the other hand, if k is too small, then the modified document vectors may be too removed from the original documents. Thus it is critical that there be some criteria for choosing the number k - this is an outstanding problem in the theory of latent semantic indexing.

There are other ways of looking at the lower rank approximation problem. For example, put

$$M = M(t \times d) = \text{the set of all } t \times d \text{ matrices} = \mathbb{R}^{td},$$

and consider

$$M_k = \text{the subset consisting of all rank at most } k \text{ matrices} \subset M.$$

Note that $M_k - M_{k-1}$ is the set of all $t \times d$ matrices of rank exactly k. Several observations are in order. To begin with, the sets $M_k \subset M = \mathbb{R}^{td}$ are homogeneous affine varieties as they are defined by setting all $k \times k$ minors equal to zero. The set $M_k - M_{k-1}$ is a *generic subset* of M_k - in particular, it is open and dense in M_k - as M_{k-1} is a Zariski subset. From these considerations we obtain the following theorem.

THEOREM 5. *The minimum Euclidean distance between an element $X \in M_r - M_{r-1}$ and the subset M_k is attained along a subvariety Z_k intersecting $M_k - M_{k-1}$.*

Note that $X_k \in Z_k \cap (M_k - M_{k-1})$. It would be interesting to analyze the structure of $Z_k \cap (M_k - M_{k-1})$ and compare its elements with the k-th singular approximation of X.

3. Linear relevance feedback

Suppose we are given a collection of documents $\Delta = \{D_i,\ 1 \leq i \leq d\}$. Further suppose that the user is searching for documents that are relevant to a fixed query q. Given two documents the user may prefer one

document over the other, or may decide that he is unable to make the comparison between the two. We are discounting the possibility that the user will find the two documents equally relevant. We do this primarily for the sake of expositional clarity. After all, it is not difficult to incorporate such possibilities into our theory by introducing *equivalence classes* of documents. These considerations lead to the following definition:

DEFINITION 6. Given a document collection $\Delta = \{D_i\}$ a user preference (associated with a query q) is a partial order $\prec$ on Δ so that

$$D_1 \prec D_2 \text{ means that the user prefers } D_1 \text{ to } D_2.$$

Thus a user preference relation makes the document collection Δ into a *poset* $(\Delta, \prec)$. Given such a poset $(\Delta, \prec)$ a *document ranking function* is any map

$$\rho : \Delta \longrightarrow \mathbb{R}$$

such that

$$\rho(D_1) < \rho(D_2) \text{ whenever } D_1 \prec D_2.$$

Document ranking functions always exist: one merely goes through each linear subchain in $(\Delta, \prec)$ and assign values monotonically within the linear chain. However, we would like to work with *linear* document ranking functions, if at all possible.

DEFINITION 7. Given a document collection $(\Delta, \prec)$, a linear document ranking function is a document ranking function of the form

$$\rho = f|_\Delta ,$$

where $f : \mathbb{R}^t \longrightarrow \mathbb{R}$ is a linear map.

Recall that any linear map $\mathbb{R}^t \longrightarrow \mathbb{R}$ is of the form

$$x = (x^i) \longmapsto \sum_i a_i x^i$$

for some constant vector $a = (a_i) \in \mathbb{R}^t$. Thus we may write the map ρ as

$$\rho_a(D) = a \cdot D, \quad D \in \Delta$$

for some fixed vector $a \in \mathbb{R}^t$.

The problem with coming up with a linear document ranking function (or for that matter, any document ranking function) is that in practice the user preference is never completely known to the retrieval system as the user is unable to examine all the documents - or even a significant number - in the document collection previous to the search. The idea behind the method of *relevance feedback* is to use a recursive process that goes something like this:

1. At each step a document ranking function which is consistent with a sample set of documents (i.e., a training set where the user specifies his preference fully) is found;
2. More documents are retrieved using the document ranking function thus obtained, and the retrieved documents become the new training set;
3. The process continues until the user is satisfied with the documents retrieved.

In linear relevance feedback we would want to find a linear document ranking function at each step of the above recursive process. Indeed following [Wong-Yao] we now describe an algorithm where the document ranking function is given by $\rho_q(D) = q \cdot D$ with q a modified query vector.

DEFINITION 8. Suppose $\Delta_1 \subset \Delta$ is a subset of documents, where the user preference is fully known. Then a vector $q \in \mathbb{R}^t$ is called a query vector associated with Δ_1 if ρ_q is a document ranking function when restricted to Δ_1.

An important problem in the theory of linear relevance feedback is to find natural criteria under which there exists a query vector associated with an arbitrary subset Δ_1 of Δ. Given a fixed subset Δ_1 however, the existence (and uniqueness) problem for the associated query vectors is not a difficult problem. To see this, put

$$B_1 = \{b = D' - D : D', D \in \Delta_1,\ D \prec D'\} \subset \mathbb{R}^t.$$

The following result is easy.

LEMMA 9. *A vector $q \in \mathbb{R}^t$ is a query vector associated with Δ_1 if and only if*

$$(*) \qquad q \cdot b > 0 \text{ for every } b \in B_1.$$

Note that $(*)$ is a system of linear inequalities on $\mathbb{R}^t$; as such the solutions define a convex region in $\mathbb{R}^t$, possibly empty. Supposing that this system is consistent we now describe an algorithm for finding a solution:

1. Set $q_k = q_0$ (q_0 may be given by the user).
2. Set $B_1^k = \{b = D' - D : D', D \in \Delta_1,\ D \prec D' \text{ and } q_k \cdot b \leq 0\}$.
3. If $B_1^k = \emptyset$, then set $q = q_k$ and stop.
4. Set $q_{k+1} = q_k + \sum_{b \in B_1^k} b$.
5. Set $k = k + 1$ and go back to Step 2.

It would be interesting to find natural conditions under which the above algorithm terminates in finitely many steps. We caution the reader that in [Wong-Yao] the notion of a user preference is defined somewhat differently. Essentially, a user preference for them is a linear chain with equivalence classes, which, we feel, may be too restrictive. They then impose an extra condition on the user preference called *weak linearity* to guarantee that the above algorithm terminates in finitely many steps.

On occasion it may be necessary to consider higher order document ranking functions.

DEFINITION 10. Given $(\Delta, \prec)$ an n-th order document ranking function is a map

$$\lambda : \Delta \subset \mathbb{R}^t \longrightarrow \mathbb{R}$$

given as a restriction to Δ of an n-th order polynomial (not necessarily homogeneous but without a constant term) in t variables.

More explicitly, we can write

$$\lambda(x) = \sum_{|J|=1}^{n} a_J x^J, \quad x = (x^j) \in \mathbb{R}^t,$$

where J is a multi-index of order $|J|$ and not all the leading degree terms are zero. We conjecture that given any document collection $(\Delta, \prec)$ there always is a document ranking function of sufficiently high order associated with it.

References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, *Indexing by latent semantic analysis*, Journal of the American Society for Information Science **41** (1990), no. 6, 391-407.

[2] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[3] S. K. M. Wong and Y. Y. Yao, *Query formulation in linear retrieval models*, Journal of the American Society for Information Science **41** (1990), no. 5, 334-341.

Department of Mathematics
University of Texas - Pan American
Edinburg, TX 78539
E-mail: kyang@math.panam.edu