To the Graduate Council:

I am submitting herewith a dissertation written by Eric Peiqing Jiang entitled "Information Retrieval and Filtering Using the Riemannian SVD". I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Michael W. Berry

Michael W. Berry, Major Professor

We have read this dissertation
and recommend its acceptance:

[illegible]

Jack Dongarra

Padma Raghavan

Accepted for the Council:

[illegible]

Associate Vice Chancellor
and Dean of the Graduate School

Information Retrieval and Filtering Using the Riemannian SVD

A Dissertation

Presented for the

Doctor of Philosophy Degree

The University of Tennessee, Knoxville

Eric Peiqing Jiang

August 1998

Acknowledgments

I thank my major professor, Dr. Michael Berry, for his support and guidance throughout this entire dissertation project. I wish to express my sincere appreciation for his kindness and encouragement. I also thank Dr. Charles Collins, Dr. Jack Dongarra and Dr. Padma Raghavan for serving on my dissertation committee.

I am very grateful to my wife, Alice, and my son, George, for their love, patience and understanding, and without their full support, the completion of this work would not have been possible. Finally, many thanks go to my parents for their unfailing love and for believing I would succeed even at my most doubtful moments.

This research was supported in part by the National Science Foundation under Grant No. CDA-95-29459.

Abstract

Latent Semantic Indexing (LSI) is an SVD-based conceptual retrieval technique which employs a rank-reduced model of the original (sparse) term-by-document matrix. This approach has achieved significant performance improvements over traditional lexical searching methods. With current LSI implementations, however, the ability to overcome polysemy problems (multiple meanings for a word or words) has been lacking. The Riemannian SVD (R-SVD) is a recent nonlinear generalization of the SVD which has been used for applications in systems and control. Updating LSI models based on user feedback can be accomplished using constraints modeled by the R-SVD of a low-rank approximation to the original term-by-document matrix. This dissertation presents the formulation, implementation and performance analysis of a new LSI model (RSVD-LSI) which is equipped with an effective information filtering mechanism. Two iterative algorithms for computing the related R-SVD are proposed. Experiments have shown that the RSVD-LSI model provides an efficient and robust information retrieval/filtering technique and demonstrates a new approach of updating LSI and similar vector-space models to circumvent polysemy problems and improve retrieval performance. The nonlinear filtering mechanism in RSVD-LSI may also have potential applications in designing certain control and security systems for information retrieval from large collections.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The growth of the Internet and the availability of enormous volumes of digital data have necessitated effective and robust techniques to retrieve and/or filter information[1]. The Internet has over 350 million pages of data and is expected to reach over one billion pages by the year 2000 [Kow97]. Webpages provide both valuable sources to solve problems as well as a huge quantity of data the average person does not care about. Digital libraries are also emerging around the world, crossing all disciplines and media and offering new levels of access to a broader audience of users and new opportunities for global exchange and understanding.

Traditionally, information is retrieved by literally matching terms in a user's query with terms extracted from documents within a collection. Lexical-based

[1]Information filtering is a problem that is closely related to information retrieval. In information filtering applications, streams of information are constantly received and matched against a user's long-term interest or profile [BDO95].

information retrieval techniques, however, can be incomplete and inaccurate due to the variability in word usage. On one side, people typically use different words to describe the same concept (*synonymy*) so that the literal terms in a user's query may be inadequate to retrieve relevant documents. On the other side, many words can have multiple meanings (*polysemy*) so that searching based on terms in a user's query could retrieve documents out of context.

Latent Semantic Indexing (LSI) attempts to circumvent the problems of lexical matching approaches by using statistically derived conceptual indices rather than individual literal terms for retrieval [BDO95, DDF+90]. LSI assumes that an underlying semantic structure of word usage exists in a document collection and uses this to estimate the semantic content of documents in the collection. This estimation is accomplished through a rank-reduced term-by-document space via the singular value decomposition (SVD) [GL96]. Using LSI, a search is based on semantic content rather than the literal terms of documents so that relevant documents which do not share any common terms with the query are retrieved.

As a vector-space retrieval technique, LSI has outperformed the lexical searching techniques quite significantly [Dum91] (see Section 4.1). However, in order to improve the viability and versatility of current LSI implementations, a mechanism to incorporate nonlinear perturbations to the existing rank-reduced model has been lacking.

Recently, an interesting nonlinear generalization of the singular value decomposition (SVD), referred to as the *Riemannian* SVD (R-SVD), has been proposed by De Moor [Moo93] for solving the structured total least squares problems. The R-SVD has numerous applications in systems and control.

The R-SVD can be generalized for low-rank matrices and therefore used to formulate a *new* LSI implementation for information retrieval. Updating LSI models based on user feedback can be accomplished using constraints modeled by the R-SVD of a low-rank approximation to the original term-by-document matrix. This dissertation presents the formulation, implementation and performance analysis of the new enhanced LSI model (RSVD-LSI) which is equipped with an effective information filtering mechanism. Two iterative algorithms for computing the R-SVD are proposed and one of them, Algorithm RSVD-LR2, serves as a fundamental tool in developing the RSVD-LSI model. Experiments have shown that RSVD-LSI provides an efficient information retrieval/filtering technique and demonstrates a new method of updating LSI (and similar vector-space methods) which could not be represented by traditional matrix decomposition tools such as the SVD. This new filtering technique may also have potential applications in designing certain control or security systems for information retrieval from large collections.

It has been noted that a method known as *structured total least norm* (STLN) has recently been proposed [RPG96, HPR96] for solving total least squares problems. This approach preserves any *affine* structure of a matrix A or $[A \mid b]$ and also has an attractive application in computing the best low-rank structure-preserving approximation to a given matrix. The STLN method could be used in updating LSI models. However, the required computational costs of current STLN algorithms are too high for any practical implementation for indexing large text collections. The remaining chapters of this dissertation are outlined below.

Chapter 2 provides an overview of the conventional SVD and the Riemannian SVD. In Chapter 3, the formulation of the LSI-motivated R-SVD for a low-rank matrix is described, and new iterative algorithms for solving these R-SVD problems are proposed. The implementation of Algorithm RSVD-LR2 and the formulation of the RSVD-LSI model for information retrieval and filtering are presented in Chapter 4. Chapter 5 evaluates the retrieval performance of the RSVD-LSI model on a benchmark test collection - MEDLINE. Finally, a summary of the usefulness and effectiveness of the RSVD-LSI approach and future work are discussed in Chapter 6.

Chapter 2

The Riemannian Singular Value Decomposition

2.1 Singular Value Decomposition (SVD)

The singular value decomposition (SVD) [GL96, Ber92] is a well-known theoretical and numerical tool. It defines a diagonalization of a matrix based on an orthogonal equivalence transformation. Specifically, given a rectangular m by $n\,(m \gg n)$ matrix A of rank r, the SVD of A can be defined as

$$A = U\Sigma V^T, \tag{2.1}$$

where $U = (u_1, \ldots, u_m)$ and $V = (v_1, \ldots, v_n)$ are unitary matrices and $\Sigma = diag(\sigma_1, \sigma_2, \ \ldots, \sigma_n), \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$. The σ_i's are called the singular values of A, and the u_i's and v_i's are, respectively, the left and right singular vectors associated with $\sigma_i, i = 1, \ldots, r$, and the ith singular triplet of A is defined as $\{u_i, \sigma_i, v_i\}$. The matrix A can also be expressed as a sum of r rank-one matrices:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{2.2}$$

Further, for any unitarily invariant norm $\|.\|$, the Eckart-Young-Mirsky formula [EY36, Mir60]

$$\min_{rank(B) \leq k} \|A - B\| = \|A - A_k\|, \tag{2.3}$$

where $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, provides an elegant way to produce the *best* (in a least squares sense) rank-reduced approximation to A. This approximation, however, usually differs from the original matrix in all entries. Such differences may be inappropriate for applications where particular entries may or may not be changed (see Section 3.2).

The SVD is an important computational tool used in numerous applications ranging from signal processing, control theory to pattern recognition and time-series analysis. The *total least squares* (TLS) method [HV91] is a useful data fitting technique for solving an over-determined system of linear equations $Ax \approx b$, and the most common TLS algorithm is based on the SVD of augmented matrices such as $[A \mid b]$. In fact, the TLS problem can be reduced to an equivalent minimization problem of finding a rank-deficient matrix approximation B in the Frobenius norm to A, i.e.,

$$\min_{B \in \mathcal{R}^{m \times n}, y \in \mathcal{R}^n} \|A - B\|_F, \quad \text{subject to } B\,y = 0,\ y^T y = 1. \tag{2.4}$$

By using the method of Lagrange multipliers it can be shown that the solution of (Eq. 2.4) satisfies the set of equations

$$\begin{aligned} Ay &= x\tau, \quad x^T x = 1, \\ A^T x &= y\tau, \quad y^T y = 1, \end{aligned} \tag{2.5}$$

with $\|A - B\|_F = \tau$. Hence, (x, τ, y) is the singular triplet of A corresponding to its smallest singular value.

For applications in which the matrix A has a special structure (e.g., Toeplitz or block-Toeplitz), the SVD-based TLS methods may not always be appropriate, since they do not necessarily preserve any special structure. As an extension of the TLS problem to incorporate the algebraic pattern of errors in the matrix A, *structured total least squares* (STLS) problems have been formulated. Motivated by the derivation from (Eq. 2.4) to (Eq. 2.5) and applications in systems and control, De Moor [Moo93] has proposed an interesting nonlinear generalization of SVD, referred to as the *Riemannian* SVD (R-SVD), for solving STLS problems.

The STLS has numerous applications in signal processing, system identification and control system design and analysis.

2.2 Riemannian SVD for a Full-Rank Matrix

Let $B(b) = B_0 + b_1 B_1 + \cdots + b_q B_q$ be an affine matrix function of b_i and B_i, where $b = (b_1, b_2, \ldots, b_q)^T \in \mathcal{R}^q$ is a parameter vector and $B_i \in R^{m\times n}$, $i = 0, \ldots, q$, are fixed basis matrices. Examples of affine matrix functions are structured matrices such as Toeplitz, Hankel, circulant matrices or matrices with certain zero patterns. Further, let $a = (a_1, a_2, \ldots, a_q)^T \in \mathcal{R}^q$ be a data vector, and $w = (w_1, w_2, \ldots, w_q)^T \in \mathcal{R}^q$ be a given weighting vector, then the STLS problem, that often occurs in systems and control applications, can be formulated as

$$\min_{b\in\mathcal{R}^q, y\in\mathcal{R}^n} [a,\, b,\, w]_2^2 \quad \text{subject to } B(b)\,y = 0,\ y^T y = 1. \tag{2.6}$$

Consider the STLS problem (Eq. 2.6) with the quadratic criterion

$$[a,\, b,\, w]_2^2 = \sum_{i=1}^{q} (a_i - b_i)^2,$$

where, for the sake of simplicity, the weights (w) are ignored (the general case with weights can be treated very similarly). By using the method of Lagrange multipliers, this minimization problem (Eq. 2.6) can be formulated as an equivalent Riemannian SVD problem for the matrix $A = B_0 + \sum_{i=1}^{q} a_i B_i \in \mathcal{R}^{m \times n}$. This formulation is addressed in the following theorem from [Moo93].

Theorem (STLS to R-SVD) *Consider the STLS problem*

$$\min_{b \in \mathcal{R}^q, y \in \mathcal{R}^n} \sum_{i=1}^{q} (a_i - b_i)^2 \quad \textit{subject to } B(b)\, y = 0,\ y^T y = 1, \tag{2.7}$$

where a_i, $i = 1, \ldots, q$, *are the components of the data vector* $a \in R^q$, *and* $B(b) = B_0 + b_1 B_1 + \cdots + b_q B_q$, *with* $B_i \in R^{m \times n}$, $i = 0, \ldots, q$, *a given set of fixed basis matrices. Then the solution to* (Eq. 2.7) *is given as follows:*

(a) *Find the triplet* (u, τ, v) *corresponding to the minimal* τ *that satisfies*

$$\begin{aligned} Av &= D_v u \tau, \quad u^T D_v u = 1, \\ A^T u &= D_u v \tau, \quad v^T D_u v = 1, \end{aligned} \tag{2.8}$$

where $A = B_0 + \sum_{i=1}^{q} a_i B_i$. *Here* D_u *and* D_v *are defined by*

$$\sum_{i=1}^{q} B_i^T (u^T B_i v)\, u = D_u v$$

and

$$\sum_{i=1}^{q} B_i (u^T B_i v)\, v = D_v u,$$

respectively. D_u and D_v are symmetric positive or nonnegative definite matrices[1], *and the elements of D_u and D_v are quadratic in the components of the left and right singular vectors u and v, respectively, of the matrix A corresponding to the singular value τ.*

(b) *The vector y is computed by $y = v/\|v\|$.*

(c) *The components of b, b_k, are obtained by*

$$b_k = a_k - (u^T B_k v)\tau, \ k = 1, \ldots, q.$$

Proof of Theorem: The method of Lagrange multipliers is used in this proof. Let $l \in \mathcal{R}^m$ be a vector of Lagrange multipliers and $\lambda \in \mathcal{R}$ is a scalar Lagrange multiplier, then the Lagrangian for (Eq. 2.7) is defined by

$$\mathcal{L}(b, y, l, \lambda) = \sum_{i=1}^{q}(a_i - b_i)^2 + l^T(B_0 + \sum_{i=1}^{q} b_i B_i)y + \lambda(1 - y^T y).$$

Setting all derivatives equal to zero (and absorbing irrelevant constants in the Lagrange multipliers) gives

$$a_k - b_k = l^T B_k y, \ (k = 1, 2, \ldots, q)$$

[1]The Riemannian SVD of A can be interpreted as a *restricted* SVD (RSVD) [Zha91] with the Riemannian metrics (D_u and D_v) in the column and row space of A.

$$(B_0 + \sum_{i=1}^{q} b_i B_i)y = 0, \tag{2.9}$$

$$(B_0^T + \sum_{i=1}^{q} b_i B_i^T)l = y\lambda,$$

$$y^T y = 1,$$

and from $l^T B(b)y = 0$ it is easy to see that $\lambda = 0$.

Next, the relations $b_k = a_k - l^T B_k y$ in (Eq. 2.9) are used to eliminate the parameter vector b in the two middle equations of (Eq. 2.9) and yields

$$(B_0 + \sum_{i=1}^{q} a_i B_i)y = \sum_{i=1}^{q}(B_i y)(l^T B_i y), \tag{2.10}$$

$$(B_0^T + \sum_{i=1}^{q} a_i B_i^T)l = \sum_{i=1}^{q}(B_i^T l)(l^T B_i y). \tag{2.11}$$

The terms on the right-hand side of (Eq. 2.10) and (Eq. 2.11) can be examined further to explore their special structures. Without loss of generality, let's consider the first term on the right-hand side of (Eq. 2.10), $(B_1 y)(l^T B_1 y)$, and define $B_1 = [\beta_{ij}]_{m\times n}$ and its ith row as $\bar{b}_i^T$. Then, the kth element of $(B_1 y)(l^T B_1 y)$ can be written as

$$\begin{aligned} ((B_1 y)(l^T B_1 y))_k &= \sum_{j=1}^{n} \beta_{kj} y_j \sum_{r=1}^{m}\sum_{s=1}^{n} \beta_{rs} l_r y_s, \\ &= \sum_{r=1}^{m} (y^T \bar{b}_k)(\bar{b}_r^T y) l_r. \end{aligned}$$

Hence,

$$(B_1y)(l^TB_1y) = \begin{pmatrix} y^T\bar{b}_1 \\ y^T\bar{b}_2 \\ \vdots \\ y^T\bar{b}_m \end{pmatrix} \begin{pmatrix} \bar{b}_1^Ty & \bar{b}_2^Ty & \cdots & \bar{b}_m^Ty \end{pmatrix} l.$$

In other words, the term $(B_1y)(l^TB_1y)$ can be expressed as a product of a rank-one (nonnegative definite) matrix and the vector l. In general, this derivation can be repeated for other terms of (Eq. 2.10) and then the right-hand side of (Eq. 2.10) can be written as

$$\sum_{i=1}^{q}(B_iy)(l^TB_iy) = D_yl, \tag{2.12}$$

where D_y is a symmetric matrix which is a sum of q rank-one nonnegative definite matrices and hence, D_y itself is nonnegative or positive definite. The elements of D_y are quadratic functions of the components of y. A similar process can be applied to the right-hand side of (Eq. 2.11) and yields

$$\sum_{i=1}^{q}(B_i^Tl)(l^TB_iy) = D_ly, \tag{2.13}$$

where D_l is a symmetric nonnegative or positive definite matrix, with elements that are quadratic functions of the components of l.

Now, define $x = l/\|l\|$ and $\sigma = \|l\|$. Then $l = \sigma x$, and define a matrix D_x from D_l by replacing every component of l in D_l by its corresponding component in

x. Since the elements of D_l are quadratic in the components of l, it follows that $D_l = D_x\sigma^2$. If $A \in \mathcal{R}^{m\times n}$ is defined as $A = B_0 + \sum_{i=1}^{q} a_i B_i$, then (Eqs. 2.10-2.13) yield

$$\begin{aligned} Ay &= D_y x\sigma, \quad x^T x = 1, \\ A^T x &= D_x y\sigma, \quad y^T y = 1. \end{aligned} \tag{2.14}$$

From (Eq. 2.14) it follows directly that

$$x^T A y = y^T A^T x = x^T D_y x\sigma = y^T D_x y\sigma. \tag{2.15}$$

Using (Eqs. 2.9-2.10) and (Eq. 2.15), the original quadratic criterion in (Eq. 2.7) can be written as

$$\begin{aligned} \sum_{i=1}^{q}(a_i - b_i)^2 &= \sum_{i=1}^{q}(l^T B_i y)^2 \\ &= \sum_{i=1}^{q}(x^T B_i y)^2\sigma^2 \\ &= x^T \sum_{i=1}^{q}(B_i y(x^T B_i y))\sigma^2 \\ &= x^T D_y x\sigma^2 \\ &= x^T A y\sigma. \end{aligned} \tag{2.16}$$

If the triplet (u, τ, v) is the solution to (Eq. 2.8), then

$$u^T A v = (u^T D_v u)\tau = \tau. \tag{2.17}$$

By normalizing u and v, and recalling that D_u and D_v are quadratic in the components of u and v, respectively, it follows that

$$\begin{aligned} A\frac{v}{\|v\|} &= \frac{D_v}{\|v\|^2}\frac{u}{\|u\|}(\tau\|u\|\|v\|), \\ A^T\frac{u}{\|u\|} &= \frac{D_u}{\|u\|^2}\frac{v}{\|v\|}(\tau\|u\|\|v\|). \end{aligned}$$

Define $x = u/\|u\|$, $y = v/\|v\|$, and $\sigma = \tau\|u\|\|v\|$. From (Eqs. 2.16 and 2.17),

$$\begin{aligned} \sum_{i=1}^{q}(a_i - b_i)^2 &= x^T A y \sigma \\ &= \frac{u^T}{\|u\|}A\frac{v}{\|v\|}(\tau\|u\|\|v\|) \\ &= \frac{u^T}{\|u\|}\frac{D_v}{\|v^2\|}\frac{u}{\|u\|}(\tau\|u\|\|v\|)^2 \\ &= \tau^2, \end{aligned}$$

which implies that the minimal τ is needed to generate a solution to the STLS problem (Eq. 2.7).

Finally, from (Eq. 2.9) the components of b, b_k, are obtained by

$$\begin{aligned} b_k &= a_k - \frac{u^T}{\|u\|} B_k \frac{v}{\|v\|} \sigma \\ &= a_k - (u^T B_k v)\tau, \quad k = 1, \ldots, q. \end{aligned}$$

□

In order to understand the R-SVD as posed in the above theorem, a few comments are needed. First, Part (c) of the theorem implies that $B = A - f(u, \tau, v)$, where f is a multilinear function of the singular triplet (u, τ, v). Second, it is interesting to notice the similarity between (Eqs. 2.4 and 2.5) and (Eqs. 2.7 and 2.8). In fact, when D_u and D_v are identity matrices, the Riemannian SVD of the matrix A will reduce to the traditional SVD of A. Lastly, it has been noted that as a nonlinear extension of SVD the R-SVD may not have a complete decomposition with $\min(m, n)$ unique singular triplets, nor the dyadic decomposition as defined in (Eq. 2.2). Also, the solution to (Eq. 2.7) may not be unique [Moo93].

2.3 An Inverse Iteration Algorithm

It is known that if both D_u and D_v are constant matrices that are independent of u and v, then the solution of (Eq. 2.8) can be obtained by computing the minimal eigenvalue (τ) via inverse iteration methods [GL96]. Based on this observation an inverse iteration algorithm for computing the R-SVD (Eq. 2.14) (or (Eq. 2.8)) of a *full-rank* matrix A has been proposed in [Moo93]. Essentially, for given matrices D_u and D_v, this algorithm performs one step of inverse iteration (using the QR decomposition of A) to obtain new estimates of the singular vectors u and v. These new vectors u and v are then used in updating matrices D_u and D_v.

Given a full-rank matrix $A \in \mathcal{R}^{m \times n}$, consider its QR decomposition

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ O \end{pmatrix},$$

where $Q_1 \in \mathcal{R}^{m \times n}$, and $Q_2 \in \mathcal{R}^{m \times (m-n)}$ are orthogonal matrices, and $R \in \mathcal{R}^{n \times n}$ is an upper triangular matrix. Then, the inverse iteration algorithm for computing the R-SVD of a full-rank matrix (Eq. 2.14) can be summarized as follows:

Algorithm RSVD-FR

1. Set $i = 0$
2. Initialize $u^{(0)}$, $v^{(0)}$ and $\sigma^{(0)}$
3. Compute matrices $D_x^{(0)}$ and $D_y^{(0)}$
4. While (not converged) do
    5. Set $i = i + 1$
    6. $z^{(i)} = R^{-T} D_x^{(i-1)} y^{(i-1)}$
    7. $w^{(i)} = -(Q_2^T D_y^{(i-1)} Q_2)^{-1}$ $(Q_2^T D_y^{(i-1)} Q_1) z^{(i)}$
    8. $x^{(i)} = Q_1 z^{(i)} + Q_2 w^{(i)}$
    9. $x^{(i)} = x^{(i)} / \|x^{(i)}\|$
    10. $y^{(i)} = R^{-1} Q_1^T D_y^{(i-1)} x^{(i)}$
    11. $\sigma^{(i)} = \|y^{(i)}\|$
    12. $y^{(i)} = y^{(i)} / \sigma^{(i)}$
    13. Compute $D_x^{(i)}$
    14. Compute $D_y^{(i)}$
    15. Convergence Test

Applications of this algorithm for solving problems in structured and weighted total least squares (TLS), noisy realization, system identification, and parameter estimation have been illustrated in [Moo93]. However, experiments have shown that this particular inverse iteration scheme lacks robustness for computing the R-SVD of arbitrary matrices in (Eq. 2.14) due to potential breakdowns associated with singular iteration matrices. Algorithm RSVD-FR can be modified to solve STLS problems of the form in (Eq. 2.8).

Chapter 3

The Riemannian SVD for a Low-Rank Matrix

In this chapter, the Riemannian SVD (R-SVD) of a low-rank matrix is formulated and two inverse iterative algorithms (RSVD-LR1 and RSVD-LR2) for solving R-SVD problems in information retrieval are presented. Only Algorithm RSVD-LR2, however, is used to develop a new information filtering model.

3.1 Motivation

Suppose the words *apple* and *computer* are used in a query to an IR (Information Retrieval) system. Clearly, the word *apple* can be related to different subjects and the user has a particular interest in Apple computers rather than the edible fruit. It would be preferable that the query could lead to the retrieval of only articles about Apple computers if there are any such articles in the collection. With most indexing systems available today, nevertheless, this query would return documents about Apple computers mixed with documents essentially about anything that is associated with the word *apple*. For instance, with this query, a typical WWW searching engine could mistakenly retrieve some web pages that are totally irrelevant to the Apple computers. Some of pages might be the fruit-associated such as organizations of *apple* farmers, manufacturers of *apple* juice or *apple* canning equipment; others purely word-associated such as an Internet service company (Green *Apple*, Inc.), and an university alumni regional group in New York city (Big *Apple* Chapter).

Such deficiency with these systems, i.e., failure to retrieve information that matches user interests, could be remedied significantly by implementing user relevance feedback [SB90] during the search. The success of the relevance feedback process, however, could largely depend on an appropriate local and global term weighting scheme that assigns weightings to individual terms. Also, it is assumed that all documents in the collection are formed with a conventional word usage. In information collections that involve frequent updating (with new documents and terms being added) and downdating (with obsolete documents and terms being removed) processes, the constant reassignment of weights can be computationally expensive, especially for extremely large collections.

Latent Semantic Indexing (LSI) is an SVD-based conceptual information retrieval technique which employs a rank-reduced model of the original (sparse) term-by-document space [DDF$^+$90]. It has been found that, for the LSI approach, the polysemy problems in information retrieval can be circumvented by creating a new mechanism to incorporate nonlinear perturbations to the existing rank-reduced model. Such perturbations, which can arise in the form of users feedback and interactions with LSI model, can be represented by changes in certain single or clustered entries, rows, and columns of the original term-by-document matrix used to define the rank-reduced LSI model. The updated LSI implementation using a modified R-SVD has the ability to filter out irrelevant documents by moving them to the bottom of the retrieved (ranked) document list.

3.2 Riemannian SVD for a Low-Rank Matrix

3.2.1 Formulation

The Riemannian SVD of a full-rank matrix [Moo93] (R-SVD, see Chapter 2) arising from applications in systems and control can be modified and used to formulate the Riemannian SVD of a low-rank matrix, which has applications in information retrieval. Assume A is an $m \times n \, (m \gg n)$ term-by-document matrix, and A_k is a corresponding LSI rank-k approximation of A [BDO95]. Suppose A_k

has the splitting

$$A_k = A_k^F + A_k^C, \tag{3.1}$$

where A_k^F defines the set of terms in A_k whose term-document associations are not allowed to change, and A_k^C is the complement of A_k^F.

Consider the following minimization problem:

$$\min_{B_k \in \mathcal{R}^{m\times n}, y\in\mathcal{R}^n} \|A_k - B_k\|_F^2 \quad \text{subject to } B_k y = 0,\ y^T y = 1,\ B_k = A_k^F + B_k^C. \tag{3.2}$$

Here, the matrix B_k is an approximation to the matrix A_k satisfying the constraints in (Eq. 3.2), which include preserving the specified elements of A_k (i.e., A_k^F). Using Lagrange multipliers [Moo93], this constrained minimization problem can be formulated as an equivalent R-SVD problem for the rank-k model A_k:

$$\begin{aligned} A_k v &= D_v u \tau, \quad u^T D_v u = 1, \\ A_k^T u &= D_u v \tau, \quad v^T D_u v = 1, \end{aligned} \tag{3.3}$$

where the metrics D_u and D_v are defined by

$$\begin{aligned} D_v &= \operatorname{diag}\left(P \operatorname{diag}(v)\, v\right), \\ D_u &= \operatorname{diag}\left(P^T \operatorname{diag}(u)\, u\right), \end{aligned} \tag{3.4}$$

and P is a *perturbation* matrix having zeros at all entries of A_k^F and nonzeros elsewhere.

3.2.2 Derivation

(Eqs. 3.3-3.4) can be derived from (Eq. 3.2) as follows:

Let $G = (g_{ij}) \in \mathcal{R}^{m \times n}$ be a matrix of Lagrange multipliers with $g_{ij} = 0$ when $(i, j) \in B_k^C$. Then, the Lagrange function for (Eq. 3.2) is defined by

$$\begin{aligned} \mathcal{L}(B, y, l, \lambda) &= \sum\sum_{(i,j)\in B_k^C} (a_{ij} - b_{ij})^2 + l^T B_k y + \lambda(1 - y^T y) \\ &\quad + \sum\sum_{(i,j)\in A_k^F} g_{ij}(a_{ij} - b_{ij}). \end{aligned}$$

This leads to the set of equations

$$B_k^C = A_k^C + G - l y^T,$$

$$B_k^T l = y\lambda,$$

$$B_k y = 0,$$

$$y^T y = 1.$$

Since $y^T B_k^T = 0$ and $y^T B_k^T l = y^T y \lambda$, it follows that $\lambda = 0$.

By eliminating the matrix B_k, it can be shown that

$$(A_k + G)^T l = y(l^T l),$$

$$(A_k + G)y = l,$$

$$y^T y = 1.$$

Since $B_k^C = A_k^C + G - ly^T$, it follows that

$$g_{ij} = \begin{cases} l_i y_j, & \text{if } (i,j) \in A_k^F \\ 0, & \text{otherwise} \end{cases}$$

and hence,

$$B_k^C(i,j) = \begin{cases} A_k^C(i,j), & \text{if } (i,j) \in A_k^F \\ A_k^C(i,j) - l_i y_j, & \text{otherwise} \end{cases}.$$

Define $L = \text{diag}(l)$, $Y = \text{diag}(y)$, and the perturbation matrix $P = (p_{ij})$ as

$$p_{ij} = \begin{cases} 0, & \text{if } (i,j) \in A_k^F \\ 1, & \text{otherwise} \end{cases}.$$

Let $x = l/\|l\|$, $\sigma = \|l\|$, and $D_y = \text{diag}\,(P\ \text{diag}(y)\ y)$, and $D_x = \text{diag}\left(P^T\ \text{diag}(x)\ x\right)$, then

$$B_k = A_k - LPY, \tag{3.5}$$

and hence,

$$B_k y = A_k y - LPYy = 0. \tag{3.6}$$

From (Eq. 3.6),

$$\begin{aligned} A_k y &= LPYy \\ &= L\,\mathrm{diag}\,(P\ \mathrm{diag}(y)\ y) \\ &= D_y L \ = \ D_y\, x\, \sigma. \end{aligned} \tag{3.7}$$

Similarly, since $B_k^T l = y\lambda = 0$, or $B_k^T x = 0$, it follows directly that

$$B_k^T x = A_k^T x - Y P^T L\, x = 0. \tag{3.8}$$

Then,

$$\begin{aligned} A_k^T x &= Y P^T L\, x \\ &= Y\,\mathrm{diag}\left(P^T\ \mathrm{diag}(x)\ x\right)\ \sigma \\ &= Y D_x \sigma \ = \ D_x\, y\, \sigma. \end{aligned} \tag{3.9}$$

Now, (Eq. 3.7) and (Eq. 3.9) combine to formulate the R-SVD problem for A_k:

$$A_k y \ = \ D_y x \sigma, \quad x^T x = 1, \tag{3.10}$$

$$A_k^T x \;=\; D_x y \sigma, \quad y^T y = 1,$$

where

$$D_y = \text{diag}\left(P \text{ diag}(y)\; y\right), \; and \;\; D_x = \text{diag}\left(P^T \text{ diag}(x)\; x\right).$$

With $x = u/\|u\|$, $y = v/\|v\|$ and $\sigma = \tau\|u\|\|v\|$, the above equations can be renormalized (see Section 2.2) to obtain equations as in (Eq. 3.3) and (Eq. 3.4).

Further, since

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{i=1}^{n}(A_k(i,j) - B_k(i,j))^2 p_{ij} &= \sum_{i=1}^{m}\sum_{i=1}^{n} x_i^2 y_j^2 p_{ij}\sigma^2 \\
&= x^T D_y x \sigma^2 \\
&= x^T A y \sigma \\
&= \tau^2,
\end{aligned}$$

the solution of the R-SVD problems for a low-rank matrix (Eq. 3.2) can be expressed in terms of the triplet (u, τ, v) corresponding to the minimal τ that satisfies (Eq. 3.3).

With regard to LSI applications, the solution of (Eq. 3.3) generates an updated semantic model B_k from the existing rank-reduced model A_k. In the model B_k, some of the term-document associations, which are represented by the original term-by-document matrix A, are perturbed or changed according to the splitting specified in (Eq. 3.1). This new model B_k has the same or near rank as A_k but reflects imposed (i.e., filtered) changes on term-document associations.

3.3 The RSVD-LR Algorithms

In this section, two iterative algorithms (RSVD-LR1 and RSVD-LR2) (Riemannian SVD for a Low Rank matrix) to solve the set of nonlinear equations (Eq. 3.3) are presented.

3.3.1 Algorithm RSVD-LR1

Using an inverse iteration approach, a new algorithm (RSVD-LR1) for solving the R-SVD (Eq. 3.3) with a low-rank matrix A_k can be constructed. Assume the SVD (Eq. 2.1) of the matrix A has been computed and written as

$$A = (U_k, U_{m-k})\ \mathrm{diag}(\Sigma_k, \Sigma_{n-k})\,(V_k, V_{n-k})^T,$$

where $U = (U_k, U_{m-k})$ and $V = (V_k, V_{n-k})$. Then, the updated estimates of the singular vectors u and v as well as the singular value τ from (Eq. 3.3) can be computed by the following iteration:

$$\begin{aligned}
x &= \Sigma_k^{-1} V_k^T D_u v\, \tau, \\
u^{(new)} &= [U_k - U_{m-k}\,(U_{m-k}^T D_v U_{m-k})^{-1}(U_{m-k}^T D_v U_k)]\, x, \\
s &= \Sigma_k^{-1} U_k^T D_v u^{(new)} \tau,
\end{aligned} \tag{3.11}$$

$$
\begin{aligned}
v^{(new)} &= [V_k - V_{n-k}\,(V_{n-k}^T D_u V_{n-k})^{-1}(V_{n-k}^T D_u V_k)]\, s, \\
\tau^{(new)} &= u^{(new)T} A_k v^{(new)}.
\end{aligned}
$$

Once the vectors $u^{(new)}$ and $v^{(new)}$ have been obtained, the matrices D_u and D_v can be updated by

$$
\begin{aligned}
D_u^{(new)} &= \mathrm{diag}\,(P^T\,\mathrm{diag}\,(u^{(new)})\,u^{(new)}), \\
D_v^{(new)} &= \mathrm{diag}\,(P\,\mathrm{diag}\,(v^{(new)})\,v^{(new)}).
\end{aligned}
\tag{3.12}
$$

Algorithm RSVD-LR1 can be summarized as follows:

Algorithm RSVD-LR1

1. Set $i = 0$
2. Initialize $u^{(0)}$, $v^{(0)}$ and $\tau^{(0)}$
3. Compute matrices $D_u^{(0)}$ and $D_v^{(0)}$
4. While (not converged) do
   5. Set $i = i + 1$
   6. $x^{(i)} = \Sigma_k^{-1} V_k^T D_u^{(i-1)}\, v^{(i-1)} \tau^{(i-1)}$
   7. $y^{(i)} = -(U_{m-k}^T\, D_v^{(i-1)}\, U_{m-k})^{-1}$ $(U_{m-k}^T\, D_v^{(i-1)}\, U_k)\, x$
   8. $u^{(i)} = U_k\, x^{(i)} + U_{m-k}\, y^{(i)}$
   9. $s^{(i)} = \Sigma_k^{-1} U_k^T D_v^{(i-1)}\, u^{(i)} \tau^{(i-1)}$
   10. $t^{(i)} = -(V_{n-k}^T\, D_u^{(i-1)}\, V_{n-k})^{-1}$ $(V_{n-k}^T\, D_u^{(i-1)}\, V_k)\, s$
   11. $v^{(i)} = V_k\, s^{(i)} + V_{n-k}\, t^{(i)}$
   12. $\tau^{(i)} = u^{(i)T} A_k v^{(i)}$
   13. $D_u^{(i)} = \mathrm{diag}\,(P\,\mathrm{diag}\,(v^{(i)})\, v^{(i)})$
   14. $D_v^{(i)} = \mathrm{diag}\,(P^T\,\mathrm{diag}\,(u^{(i)})\, u^{(i)})$
   15. Convergence Test

Unfortunately, Algorithm RSVD-LR1 converges only for a subset of the R-SVD problems (Eq. 3.3) corresponding to a low-rank matrix A_k. Its applicability to (Eq. 3.3) has actually been studied to some extent and some divergent cases have been identified. For instance, if the perturbation matrix P (Eq. 3.4) has a rectangular nonzero cluster with the dimension l by w, such as

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

(with $l = 3$ and $w = 2$ in this case), Algorithm RSVD-LR1 will surely fail to produce the meaningful approximate solution to (Eq. 3.3) when the dimension of the null space of A_k^T is greater than or equal to l, or when the dimension of the null space of A_k is greater than or equal to w (see Appendix A).

An analysis of the scheme (Eq. 3.11) has indicated that the failure of Algorithm RSVD-LR1 to compute the R-SVD problems of general matrices (Eq. 3.3) is mainly attributed to the singularity of the matrix $U_{m-k}^T D_v U_{m-k}$ and/or the matrix $V_{n-k}^T D_u V_{n-k}$ in (Eq. 3.11) during the iteration. The selection of nonzero elements in the perturbation matrix P has a direct and notable impact on the structures of the matrices D_u and D_v. Depending on the number and positions of these nonzero elements in P, only few or many zero diagonal entries can be produced in these two matrices and the latter cases (having numerous zero diagonal entries) would lead to singular matrices $U_{m-k}^T D_v U_{m-k}$ and $V_{n-k}^T D_u V_{n-k}$. Also, the selection of P could create unequal levels of nonlinearity in (Eq. 3.3) between the equations associated with nonzero diagonal elements of D_u and D_v and those associated with zero diagonal elements of these two matrices. A significant imbalance of nonlinearity among equations could lead to divergence or stalling behavior of the iteration. Ultimately, the various ways of selecting nonzero elements in P for applications in information retrieval would render (Eq. 3.3) as an extremely difficult set of nonlinear equations to solve.

3.3.2 Algorithm RSVD-LR2

In order to compute the R-SVD (Eq. 3.3) for information retrieval applications and circumvent the singularity problems associated with Algorithm RSVD-LR1, an improved inverse iteration algorithm (RSVD-LR2) has been developed. As in Algorithm RSVD-LR1, this algorithm updates the left singular vector u and the right singular vector v by decomposing them into two orthogonal components, and then uses these new vectors to update the matrices D_u and D_v. Specifically, it adopts the same formulas to compute intermediate vectors x and s as presented in

(Eq. 3.11), but uses a different and much simpler way to estimate the orthogonal projection of the vector u onto the null space of A^T and the vector v onto the null space of A, respectively. This new approach is effective in overcoming the singularity problems that arise in (Eq. 3.11). Current experiments have demonstrated that Algorithm RSVD-LR2 works very well for the R-SVD problems (Eq. 3.3) which arise from information retrieval/filtering applications.

Algorithm RSVD-LR2 can be summarized as follows:

Algorithm RSVD-LR2

1. Set $i = 0$
2. Initialize $u^{(0)}$, $v^{(0)}$ and $\tau^{(0)}$
3. Compute matrices $D_u^{(0)}$ and $D_v^{(0)}$
4. While (not converged) do
    5. Set $i = i + 1$
    6. $x^{(i)} = \Sigma_k^{-1} V_k^T D_u^{(i-1)} v^{(i-1)} \tau^{(i-1)}$
    7. $y^{(i)} = U_{m-k}^T u^{(i-1)}$
    8. $u^{(i)} = U_k x^{(i)} + U_{m-k} y^{(i)}$
    9. $s^{(i)} = \Sigma_k^{-1} U_k^T D_v^{(i-1)} u^{(i)} \tau^{(i-1)}$
    10. $t^{(i)} = V_{n-k}^T v^{(i-1)}$
    11. $v^{(i)} = V_k s^{(i)} + V_{n-k} t^{(i)}$
    12. $\tau^{(i)} = u^{(i)T} A_k v^{(i)}$
    13. $D_u^{(i)} = \mathrm{diag}\,(P\,\mathrm{diag}\,(v^{(i)})\, v^{(i)})$
    14. $D_v^{(i)} = \mathrm{diag}\,(P^T\,\mathrm{diag}\,(u^{(i)})\, u^{(i)})$
    15. Convergence Test

There are stopping criteria (Step 15) for Algorithm RSVD-LR2 that can be easily formulated. It can be shown from (Eq. 3.5) that

$$B_k = A_k - \mathrm{diag}(u)\, P\, \mathrm{diag}(v) \tau,$$

hence

$$B_k v = 0 \text{ and } B_k^T u = 0.$$

The last two equalities lead to a practical stopping criterion

$$\|B_k v\| + \|B_k^T u\| < tol,$$

where tol is a user-specified (small) constant. This stopping criterion has been used in the experiments presented in this work.

3.3.3 Complexity

A complexity analysis for Algorithm RSVD-LR2 has been conducted and time complexities have been measured in terms of the number of floating-point operations (flops) used.

Let m be the number of terms, n be the number of documents in a collection, and k be the dimension of the term-by-document space or factors. Then, the time complexity for Algorithm RSVD-LR2 is divided into two components: *initialization* (Step 1 to Step 3) and *iteration* (Step 5 through Step 15). In the initialization phase, the singular vectors u and v, the singular value τ, and the matrices D_u and D_v are initialized and normalized. This phase requires $4(m+n)$ flops. The time complexity for each step in the iteration phase is provided in Table 3.1.

Summing the complexities in Table 3.1 and noticing that $n \ll m$, it can be

Table 3.1: The time complexity for each step in the iteration phase of Algorithm RSVD-LR2.

Step	Complexity
5	O(1)
6	2k(3n+1)
7	2m(m-k)
8	m(2m+1)
9	k(6m+1)
10	2n(n-k)
11	n(2n+1)
12	2m(2n+1)
13	2n(m+1)
14	2m(n+1)
15	7mn+3n+2m

shown that each iteration in Algorithm RSVD-LR2 requires $O(m^2)$ flops. With regard to LSI applications, since only a small number of iterations in Algorithm RSVD-LR2 is needed to generate the new perturbed model B_k from the existing model A_k, the overall computational costs in applying the Algorithm RSVD-LR2 are modest.

Chapter 4

Applications in Information Retrieval

Algorithm RSVD-LR2 described in Chapter 3 for computing the R-SVD of a low-rank matrix has been used to formulate a more filtering-based implementation of Latent Semantic Indexing (LSI) for conceptual information retrieval.

4.1 Information retrieval

As the amount of electronic information grows at an exponential rate, it is of paramount importance to have efficient and reliable techniques to retrieve information. There are a number of information retrieval systems have been developed to avoid many of the problems that are associated with lexical matching (see Chapter

1). Some systems use information about the distribution of terms across the documents for improving retrieval. As an example, the INQUERY system [CCH92] measures the similarity between documents in a collection via the probability that is determined by an inference net.

The information retrieval model considered in this dissertation is based on the vector-space model. Vector-space models rely on the premise that the conceptual meaning of a document can be derived from the document's constituent terms. They represent the association between terms and documents from a collection in a term-by-document matrix whose rows and columns are term vectors and document vectors, respectively. The entry (i, j) in the matrix represents the importance of term i in document j (sometimes it is simply the frequency of term i in document j). A *query* is a column vector with entries representing the importance of each term contained. For retrieval purposes, the document vectors and the query vector are represented in the same term-by-document space. By measuring the similarity (via a distance or angle measurement for vectors) between the query vector and all document vectors in the space, documents with close semantic content to the query presumably will be retrieved. Clearly, by placing terms and documents in the same space, such models provide an elegant way to implement relevance feedback [SB90].

As a vector-space retrieval approach, Latent Semantic Indexing (LSI) attempts to circumvent the problems of lexical matching approaches by using statistically derived conceptual indices rather than individual literal terms for retrieval [BDO95, DDF+90]. LSI assumes that there is some underlying latent semantic structure in the association of terms with documents that is partially obscured by the variability of word choice. It employs a rank-reduced term-by-document space via the singular value decomposition (SVD), and uses this space to estimate the major associative patterns of terms and documents and to diminish the obscuring noise in word usage. Since the search is based on the semantic content rather than the literal terms of documents, this approach is capable of retrieving relevant documents even when the query and such documents do not share any common terms.

Compared to other information retrieval techniques, LSI is a fully automatic indexing method and performs surprisingly well. In terms of *precision* (the proportion of returned documents that are relevant) at different levels of *recall* (the proportion of all relevant documents in a collection that are retrieved), it has achieved up to 30% better retrieval performance over lexical searching techniques [Dum91]. However, in order to improve the viability and versatility of current LSI implementations, a mechanism to incorporate nonlinear perturbations to the existing rank-reduced model has been lacking. Updating LSI models based on user feedback can be accomplished using constraints modeled by the R-SVD of a low-rank approximation to the original term-by-document matrix (see Section 3.2).

4.2 Latent Semantic Indexing (LSI)

LSI is implemented using a rank-reduced model of the original term-by-document matrix [DDF+90]. Specifically, let A be a constructed $m \times n \, (m \gg n)$ term-by-document matrix

$$A = [a_{ij}], \tag{4.1}$$

where a_{ij} denotes the frequency in which term i occurs in document j. Usually A is very sparse since each term typically occurs only in a subset of documents. Each of the m terms in the collection is represented as a row in the matrix, while each of the n documents in the collection is represented as a column in the matrix.

In practice, a local and a global weighting may be assigned to each nonzero entry in A to indicate (as an indexing term) its relative importance within documents and its overall importance in the entire document collection, respectively. Therefore, a nonzero weighted entry of the term-by-document matrix A can be expressed as

$$a_{ij} = l_{ij} * g_i, \tag{4.2}$$

where l_{ij} is the local weighting for term i in document j and g_i is the global weighting for term i in the document collection. Various weighting schemes are discussed in [Dum91].

Once the incidence matrix A has been constructed and properly weighted, the SVD (Eq. 2.1) is used to compute a rank-k approximation ($k \ll \min(m,n)$) to A, referred to as A_k, which can be written as

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U_k \Sigma_k V_k^T, \tag{4.3}$$

where U_k and V_k, respectively, contains the first k columns of U and V (Eq. 2.1). The columns of U_k and V_k define the singular vectors which correspond to the first (largest) k singular values of A, i.e., diagonal entries of Σ_k. In other words, A_k captures the first k singular triplets of the matrix A. The Eckart-Young-Mirsky formula (Eq. 2.3) implies, with regard to LSI, that the k-dimensional term-by-document space represented by A_k is the closest rank-k approximation in a least squares sense to the original term-by-document space which is represented by A. It is assumed that the dimension reduction of the term-by-document space helps reveal the underlying semantic structure in the association of terms with documents and removes the obscuring noise in word usage.

In the reduced term-by-document space, individual terms and documents are positioned by the left (U_k) and right (V_k) singular vectors, respectively, and coordinates of the space are often scaled by the corresponding singular values (diagonal elements of Σ_k), allowing clusters of terms and documents to be easily identified. Semantically related terms and documents are assumed to be near each other in this space.

In LSI, a user's query, treated as a *pseudo-document* $\hat{q}$, is usually represented as the sum of the term vectors corresponding to the terms specified in the query scaled by the inverse of the singular values (Σ_k^{-1}) [DDF+90, LB97],

$$\hat{q} = q^T U_k \Sigma_k^{-1}, \tag{4.4}$$

where q is vector containing the weighted term-frequency counts of the terms that appear in the query, and it can be projected into the term-by-document space. The *relevance feedback* mechanism can be incorporated in a typical LSI model. Relevance feedback uses the terms in relevant documents to supplement the user's original query, allowing better retrieval performance [SB90]. Let $d \in \mathcal{R}^n$ be a vector whose nonzero elements are indices specifying the relevant document vectors. Then, a *relevance feedback query* can be constructed from the original query (Eq. 4.4) by adding the vector sum of the relevant documents identified by the user. The modified query is given by

$$\hat{q} = q^T U_k \Sigma_k^{-1} + d^T V_k. \tag{4.5}$$

Using one of several similarity measures such as the cosine measure[1], the

[1] The cosine, which measures the angle between the pseudo-document vector and the document vectors in the rank-reduced space, is merely used to rank-order documents and its explicit value is not always an adequate measure of relevance [BDO95] (see Section 4.4).

pseudo-document vector is then compared against all document vectors in the space. The terms and documents are ranked according to the results of the similarity measure used, and only those semantically related documents that presumably are the *closest* vectors are retrieved.

4.3 New LSI implementation

The new LSI implementation proposed in this work is an updated LSI model equipped with a mechanism to incorporate *nonlinear* perturbations to the existing rank-reduced model A_k. As stated previously, based on user feedback this updating process can be accomplished using constraints modeled by the R-SVD of A_k (see Section 3.2). This new implementation provides an efficient information filtering technique and demonstrates a new approach of updating LSI (and similar vector-space models) that could not be represented by traditional matrix decompositions such as the SVD.

In LSI, all indexing terms and documents in a collection as well as a user's query are represented as vectors in a rank-reduced term-by-document space. The documents are compared against the query and then ranked based on their *closeness* to the query. Due to the variability in word usage (e.g., polysemy) an LSI-based IR system may not always produce responses that accurately reflect the conceptual meaning of the user's query.

The new LSI implementation attempts to overcome these problems by performing appropriate nonlinear perturbations to the existing term-by-document model A_k in the sense that some term-document associations are changed and the corresponding term and document vectors are reconstructed and repositioned in the new perturbed rank-reduced term-by-document space. Based on user feedback or interaction with the system, a perturbation matrix P in (Eq. 3.4) can be constructed which defines the splitting of A_k as in (Eq. 3.1). For all the terms and documents in A_k, the matrix P determines the set of terms and the set of documents whose term-document associations are allowed to change (while other associations are fixed). The perturbation matrix P can also be used to have more control in ranking the returned documents which is discussed at the end of this chapter.

Once the perturbation matrix P has been constructed, the R-SVD problem for the low-rank matrix A_k (Eqs. 3.3-3.4) can be formulated. Algorithm RSVD-LR2 (proposed in Section 3.3) is then applied to compute this R-SVD, and its approximate solution generates a new perturbed term-by-document space B_k which satisfies the minimization problem in (Eq. 3.2). The new model B_k, having the same or near rank as A_k, reflects the changes in the term-document associations specified by the user in order to better capture the context of the user's query and thereby retrieve more relevant information (documents).

4.4 An example

As an example, Table 4.1 illustrates 18 general topics and associated keywords which might be found in an encyclopedia. These 18 topics can be classified into three different concepts: *Business Recession, Tectonics* and *Mental Disorders.* In this document collection, the keyword *depression* plays a special role in the sense that it is polysemous, and occurs in each of the three groups of documents. Depending on the group it occurs, the word *depression* could refer to an economic condition, the land formation or a mental state.

Corresponding to the collection of topics and keywords in Table 4.1 is the 32×18 term-by-document matrix shown in Table 4.2. In this example, the elements of the matrix are the binary-frequencies in which a keyword or term occurs in a topic or document (see Section 4.2). For instance, the terms *business, depression, tax, unemployment* (and only these four terms) occur in topic A1. For the sake of simplicity, local and global term weightings (Eq. 4.2) are ignored here.

To implement LSI, a rank-k ($k = 2$ in this case) approximation A_k of the term-by-document matrix A defined in Table 4.2 is constructed via the truncated SVD (Eq. 2.2), and the singular vectors as well as the singular values of A_k are used to position terms, documents and the query as individual vectors in the rank-reduced term-by-document space. Assume that the cosine similarity measure is applied, then all documents (and terms) are compared with the query in the space

Table 4.1: Keywords from a collection of heterogeneous articles.

Topic	Label	Keywords
Recession	A1	business, depression, tax, unemployment
Stagflation	A2	business, economy, market
Business Indicator	A3	business, market, production
Business Cycle	A4	depression, liquidation, prosperity, recovery
Unemployment	A5	compensation, unemployment, welfare
Inflation	A6	business, market, price
Welfare	A7	benefit, compensation, unemployment
Continental Drift	B1	basin, fault, submergence
Plate Tectonics	B2	depression, fault
Great Basin	B3	basin, drainage, valley
Valley	B4	depression, drainage, erosion
Lake Formation	B5	basin, drainage, volcano
Psychosis	C1	counseling, illness, mental
Alcoholism	C2	counseling, depression, emotion
Mental Health	C3	depression, rehabilitation, treatment
Depression Disorder	C4	disturbance, drug, illness
Manic Depression	C5	distractibility, illness, talkativeness
Mental Treatment	C6	counseling, drug, psychotherapy

Table 4.2: The 32×18 term-by-document matrix A corresponding to the topics/keywords listed in Table 4.1.

	Documents																	
Terms	A1	A2	A3	A4	A5	A6	A7	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5	C6
basin	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
benefit	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
business	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
compensation	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
counseling	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
depression	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	0
distractibility	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
disturbance	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
drainage	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
drug	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
economy	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
emotion	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
erosion	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
fault	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
illness	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0
liquidation	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
market	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
mental	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
price	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
production	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
prosperity	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
psychotherapy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
recovery	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rehabilitation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
submergence	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
talkativeness	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
tax	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
treatment	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
unemployment	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
valley	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
volcano	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
welfare	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

and ranked according to their (vector) cosines with the query. Table 4.3 shows the returned lists of topics and the corresponding cosine ranks from the *current* LSI implementation for the double-word queries of (*business, depression*) and (*depression, economy*). It should be assumed that all of the returned document lists presented are rank-ordered by cosines with the query vectors (see Section 4.2).

Note that the two returned document lists shown in Table 4.3 contain mixed topics across the entire collection and not all of the top-ranked documents accurately reflect or match the conceptual meanings of the queries. The polysemy of the term *depression* in the queries is the contributing factor to the suboptimal rank ordering of documents. Although implementing relevance feedback [SB90] could be one way to circumvent the difficulty under certain restrictions, the nonlinear filtering mechanism proposed in this work (see Section 4.3) could provide an efficient solution to this polysemy problem. This innovative filtering approach is based on constraints modeled by the R-SVD of the low-rank approximation A_k. By performing nonlinear perturbations to the model A_k, which can arise in the form of user's interactions/responses to the current LSI model, this approach could successfully filter out irrelevant documents by effectively lowering their similarity to the query or simply moving them farther down the returned rank-ordered document list.

Specifically, in this example, given the rank-reduced term-by-document model

Table 4.3: Returned topic lists of the *current* LSI implementation for queries *business, depression* (query1) and *depression, economy* (query2). Topics are ranked according to their cosines with the query.

Query			
query1		query2	
A1	1.00	C3	.96
A7	.99	B2	.96
A5	.99	A4	.96
C3	.77	C2	.95
B2	.76	B4	.93
A4	.75	A1	.90
C2	.74	A5	.84
B4	.71	A7	.84
A2	.65	C6	.82
A3	.65	C1	.81
A6	.52	B1	.76
C6	.52	B3	.74
C1	.50	B5	.74
B1	.43	C4	.69
B3	.40	C5	.67
B5	.40	A3	.28
C4	.34	A2	.28
C5	.31	A6	.28

A_2, the following 32×18 perturbation matrix

$$P^{(1)} = \begin{pmatrix} \vec{O}_{5\times18} \\ \\ \vec{E}^{(1)}_{1\times18} \\ \\ \vec{O}_{26\times18} \end{pmatrix}, \tag{4.6}$$

can be constructed, where $\vec{O}_{5\times18}$ and $\vec{O}_{26\times18}$ are, respectively, the 5×18 and the 26×18 zero submatrices and $\vec{E}^{(1)}_{1\times18}$ is simply the row vector

$$\vec{E}^{(1)}_{1\times18} = \begin{pmatrix} 0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1 \end{pmatrix}, \tag{4.7}$$

assuming that a user wants to see top-ranked topics about *depression* in terms of economic issues only. Algorithm RSVD-LR2 can be used to compute the approximate solution of the R-SVD defined by (Eqs. 3.3 and 3.4) for the rank-reduced model A_2. This solution leads to a new, updated low-rank term-by-document model B_2 where term associations for *depression* are perturbed and diminished in the context of land formations or mental state only and not for economic conditions. Finally, in terms of the singular vectors and the singular values of the perturbed model B_2, all topics in the collection are compared against the query and then ranked based on their relevance to the query. If the user is primarily interested in tectonics, or would like to retrieve information about mental disorders, the corresponding perturbation matrices $P^{(2)}$ and $P^{(3)}$, which contain

$$\vec{E}^{(2)}_{1\times 18} = (1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1),$$

and

$$\vec{E}^{(3)}_{1\times 18} = (1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0),$$

can be similarly defined, respectively. To satisfy different user interests, specifications for the perturbation matrix will lead to different outcomes in retrieval. The effectiveness of the R-SVD enhanced LSI implementation is demonstrated in Table 4.4 with the returned topic lists for the query *depression* and perturbation matrices $P^{(1)}$, $P^{(2)}$, and $P^{(3)}$, respectively. Notice how in each case, the topics from the appropriate context are filtered to the top of the returned topic list.

4.5 Other issues

As described in previous sections, Algorithm RSVD-LR2 for the existing rank-reduced model A_k generates an updated term-by-document model B_k where certain term associations are perturbed. The effectiveness of information filtering in B_k has been formally investigated.

4.5.1 Dimensionality of the term-by-document space

Choosing an appropriate dimension k for the rank-reduced model A_k to capture

Table 4.4: Returned topic lists of the *new* LSI implementation for the query *depression* by using different perturbation matrices. Topics are ranked according to their cosines with the query.

Perturbation Matrix					
P(1)		P(2)		P(3)	
A4	.98	B2	1.00	C2	1.00
A1	.52	B4	.98	C3	1.00
A5	.45	B1	.74	C6	.82
A7	.45	B3	.71	C1	.80
A3	.12	B5	.71	C4	.62
A2	.12	A6	.14	C5	.58
A6	.12	A3	.4	A6	.15
C5	-.95	A2	.14	A3	.15
C4	-.95	A7	-.72	A2	.15
B3	-.97	A5	-.72	A7	-.66
B5	-.97	A1	-.81	A5	-.66
B1	-.99	C5	-.92	A1	-.77
C1	-.99	C4	-.93	B3	-.94
C6	-1.00	C1	-.97	B5	-.94
C3	-1.00	C6	-.97	B1	-.95
B2	-1.00	C2	-1.00	B4	-1.00
C2	-1.00	A4	-1.00	B2	-1.00
B4	-1.00	C3	-1.00	A4	-1.00

the major term-document associations could be a key factor in success of the information retrieval, and has been studied in [DDF+90]. However, how to search for the optimal value k of the space dimension is an open problem.

In applications of the proposed LSI implementation, determining an appropriate dimension of the term-by-document space is still somewhat dataset-dependent. Thus far, experiments have suggested a relatively small k (about one-tenth of the number of documents in the dataset or smaller) for the dimension of the term-by-document space A_k, or for its perturbed model B_k, in which a clearer information filtering effect will be demonstrated. In general, a *coarse* perturbed approximation B_k to A_k, measured by a reasonable distance between A_k and B_k, would be preferred and could be a contributing factor in filtering out irrelevant information during the search. One interpretation for this coarse model is that perturbation influences on those selected terms and documents are gradually diminished as the approximate model B_k becomes more close to A_k. Therefore, only a few iterations in Algorithm RSVD-LR2 are needed to obtain the new perturbed model B_k from the given model A_k. These computationally favorable requirements of a low-dimensional term-by-document space and a small number of needed iterations of Algorithm RSVD-LR2 suggest that this new approach could be applied to large document collections.

In high-dimensional spaces ($k > 2$ in this small example), the filtering mechanism may not be as effective as demonstrated in low-dimensional spaces. Other factors such as keywords in queries and the word usage patterns within the document collection play a role as well. But experiments have shown that the filtering effectiveness deteriorates as the dimension of the space grows. Consider the returned topic lists in Table 4.5 for the same query *depression* and the same perturbation matrix $P^{(1)}$ defined in (Eq. 4.6) but using term-by-document spaces of different dimensions. Experiments suggest that in a high-dimensional space, perturbed documents are more freely allowed to move towards those unperturbed documents, which may facilitate the retrieval of mixed (relevant and irrelevant) documents.

It has been found that adding more conceptually related keywords in a query could be a way to regain the filtering effect in high-dimensional term-by-document spaces (a relevance feedback step). The other approach for a desired filtering outcome is to reduce weightings on irrelevant documents in terms of the perturbed matrix P, which is discussed in the next section.

4.5.2 Weightings on perturbed documents

In the new filtering-based LSI implementation, the perturbation matrix P ($P = (p_{i,j})$) plays an important role in marking entries of the rank-reduced term-by-document model A_k. The binary assignments in P are used to identify terms as members of sets in the splitting (Eq. 3.1) for A_k. The assignment $p_{ij} = 1$ indicates that the term association for the term i is allowed to change in the document

Table 4.5: Returned topic lists of the *new* LSI implementation for the query *depression* by using high-dimensional spaces. Topics are ranked according to their cosines with the query.

Space Dimension					
k = 3		k = 4		k = 5	
A4	.99	A4	.98	A4	1.00
A1	.55	A1	.62	A1	.55
A7	.46	A5	.40	C5	.32
A5	.46	A7	.40	C4	.28
A2	.08	C5	.07	B3	.22
A6	.08	C4	.03	B5	.22
A3	.08	A3	.01	B1	.19
B5	-.19	A2	.01	C1	.09
B3	-.19	A6	.01	A5	.04
B1	-.25	B5	-.01	A7	.04
C5	-.57	B3	-.01	A2	.02
C4	-.59	B1	-.08	A3	.02
C1	-.72	C1	-.19	A6	.02
C6	-.75	C6	-.26	C6	.02
B4	-.91	B4	-.92	B4	-.84
C2	-.98	C2	-.96	C2	-.89
B2	-.99	B2	-.99	B2	-.98
C3	-1.00	C3	-1.00	C3	-1.00

j, while $p_{ij} = 0$ simply marks no change of this term-document association. For example, the matrix $P^{(1)}$ defined in (Eqs. 4.6 and 4.7) implies that the term associations for the keyword *depression* are only allowed to change (diminish) within the first seven topics in the collection.

The assignment of positive nonzero entries (not necessarily 1) in P can also be interpreted as reciprocals of the elementwise relative weightings in the related *total least squares* problems [Moo93, HV91]. This interesting connection can be used to create a new technique with more *control* in changing term-document associations and in ranking the returned documents. In general, the larger the nonzero value (in P) is, the weaker the corresponding term-document association. Table 4.6 lists the returned topics for the query *depression* with the perturbation matrix $P^{(4)}$ and $P^{(5)}$ which has the component

$$\vec{E}^{(4)}_{1\times 18} = (0,0,0,0,0,0,0,1,1,1,1,1,5,5,5,5,5,5),$$

and

$$\vec{E}^{(5)}_{1\times 18} = (1,1,1,1,1,1,1,5,5,5,5,5,0,0,0,0,0,0),$$

respectively.

From Table 4.6 it can be seen that the magnitude of those nonzero entries in the perturbation matrix P has an important impact on rank-ordering the returned documents. In this example, the topics are ranked (as groups) according to their

Table 4.6: Returned topic lists of the *new* LSI implementation for the query *depression* by applying different weightings to perturbed documents. Topics are ranked according to their cosines with the query.

Perturbation Matrix			
P(4)		P(5)	
A4	1.00	C2	1.00
A1	.61	C3	.99
A7	.52	C6	.92
A5	.52	C1	.91
A3	.06	C4	.77
A2	.06	C5	.76
A6	.06	A4	.74
B2	-.04	A1	.15
B3	-.11	A7	.14
B5	-.11	A5	.14
B1	-.11	A6	.09
B4	-.32	A3	.09
C4	-.98	A2	.09
C1	-.98	B3	-.99
C1	-.99	B5	-.99
C6	-1.00	B1	-.99
C3	-1.00	B2	-1.00
C2	-1.00	B4	-1.00

corresponding values in P which are associated with the term *depression*. A larger (positive) nonzero entry in P specifies a weaker term-document association or at least an association which may be perturbed or diminished.

Chapter 5

Retrieval Performance

The new LSI model described in Sections 4.3 and 4.4 has been implemented in MATLAB Version 5. Experiments have shown that this updated implementation provides an effective conceptual information retrieval/filtering technique. The focus of this chapter is to evaluate the performance of this new LSI implementation by comparing it with the traditional LSI approach with and without *relevance feedback*. Characteristics of the MEDLINE document collection used in testing, standard performance measures, and a few testing parameter choices are also discussed.

5.1 MEDLINE

The MEDLINE collection is a commonly studied set of medical abstracts from the National Library of Medicine [DDF+90]. The MEDLINE collection contains

1033 documents, a set of 1033 corresponding document titles, and a set of 30 queries as well as *relevance judgments* for each query. The relevance judgments are lists of documents in the collection relevant to each query. As a preprocessing step, a total of 439 common words such as *the* or *therefore* were discarded from the set of possible keywords (terms) for this collection. These common words were used to form a *stop list* [1] for the term indexing process. Any term that occurs in more than one document and is not on the stop list was used as an indexing term. After keyword extraction, a sparse term-by-document matrix A of dimensions 5831 by 1033 was produced. Characteristics of the MEDLINE collection are listed in Table 5.1.

5.2 Three LSI implementations

The original or current LSI model (C-LSI) [DDF+90], and LSI with relevance feedback (RF-LSI) [Dum91] as well as the new LSI implementation based on R-SVD (RSVD-LSI) are the three LSI implementations to be compared.

LSI typically uses both a local and global weighting scheme to emphasize or deemphasize the relative importance of terms within documents and the overall importance of terms across the entire document collection, respectively (see Section 4.2). For the MEDLINE test, each of the three LSI implementations used

[1] A stop list is a list of words considered to have no indexing value, used to eliminate potential indexing terms [FBY92]

Table 5.1: Characteristics of the MEDLINE collection.

Number of Documents	1033
Number of Terms	5831
Number of Queries	30
Average Number of Terms per Document	50
Average Number of Document per Term	9
Average Number of Terms per Query	10
Average Number of Documents Relevant to a Query	23
Percent Nonzero Entries in Matrix	0.86

the log-entropy weighting scheme. More specifically, let f_{ij} be the frequency of term i in document j, f_i be the global frequency of term i, and n_d be the number of documents in the collection, then the local weighting for term i in document j, l_{ij}, and the global weighting for term i in the entire document collection, g_i, of (Eq. 4.2) are given by

$$l_{ij} = \log(f_{ij} + 1),$$

and

$$g_i = 1 + \sum_j \frac{p_{ij} \log(p_{ij})}{\log(n_d)},$$

where

$$p_{ij} = \frac{f_{ij}}{f_i}.$$

Both C-LSI and RF-LSI models have been described in detail in Section 4.2. RF-LSI is essentially the same implementation as C-LSI with the exception that it uses information about returned relevant documents to supplement the initial user query and to improve the retrieval performance. RSVD-LSI is the new LSI implementation discussed in Section 4.3. Briefly, given a rank-reduced term-by-document space A_k, RSVD-LSI generates a new perturbed term-by-document space B_k where a better retrieval outcome is expected. It has been noted that while the original term-by-document matrix A is very sparse, the rank-reduced model B_k in general is a dense matrix. But fortunately, it has been observed that many of those nonzero entries in B_k are small or tiny, and these entries should have little impact on the retrieval performance of the model B_k. In this experiment with MEDLINE, any entry of B_k with its absolute value less than 10^{-2} is ignored or zeroed. This trick performed on B_k is quite successful and efficient in terms of making B_k a sparse matrix and reducing the storage and computational costs.

5.3 Performance measures

The performance of information retrieval systems is often measured in terms of two factors: *precision* and *recall.* Given a query, an information retrieval system returns a ranked list of documents. Let r_i be the number of relevant documents among the top i documents in the returned list, and r_q be the number of relevant documents to the query in the collection. The precision for the top i documents, $\mathcal{P}_i$, is then defined as

$$\mathcal{P}_i = \frac{r_i}{i}, \tag{5.1}$$

i.e., the proportion of the top i documents that are relevant. The recall $\mathcal{R}_i$ is then defined as

$$\mathcal{R}_i = \frac{r_i}{r_q}, \tag{5.2}$$

i.e., the proportion of all relevant documents in a collection that are retrieved.

A precision of 100% will be maintained as long as relevant documents are retrieved. It is directly affected by the retrieval of irrelevant documents and then gradually drops to a number close to zero. Recall typically starts off close to zero and increases as long as relevant documents are retrieved until all possible relevant documents have been retrieved. It is not affected by the retrieval of irrelevant documents and thus remains at 100% once achieved. Both precision and recall take on rational values between 0 and 1.

For a collection with multiple queries, performance can be evaluated by measuring precision at several different levels of recall. It can be done for each query separately and then averaged over all queries. Other measures of retrieval effectiveness have been proposed in literature. One example is *average precision* [Har95]. The N-point interpolated average precision for a query is defined as

$$\frac{1}{N}\sum_{i=0}^{N-1}\bar{p}\left(\frac{i}{N-1}\right),$$

where

$$\bar{p}(x) \quad = \quad \max_{\frac{r_i}{r_n}\geq x} \mathcal{P}_i.$$

Typically, 11-point interpolated average precision is used. This measure can be applied to individual queries, or averaged over several queries in a collection and expressed as a percentage.

5.4 Other parameters

Two other parameters that are needed for performance testing are the *dimensionality* of the rank-reduced term-by-document space A_k (or B_k), and the *number of returned documents to be observed* for the RF-LSI and RSVD-LSI implementations.

As stated previously, determining the optimal value of k for the term-by-document space dimension is still an open problem, and can be collection-dependent. In all MEDLINE performance tests, $k = 50$ is used. For the MEDLINE collection, the average number of documents relevant to a query is 23 (see Table 5.1), but among the 30 available queries the individual numbers of relevant documents range from 11 to 39. The large differences in numbers of relevant documents for different queries suggest that a universal window size for observing all relevant returned documents (by RF-LSI and RSVD-LSI) may not be possible. For different queries, it seems more reasonable that the returned document window sizes are proportional to the actual numbers of relevant documents in the collection, in order to demonstrate and compare the retrieval effectiveness of RF-LSI and RSVD-LSI. In fact, for a given query and with an appropriate window size, the RF-LSI model can present those *relevant* documents to construct a relevance feedback query in order to facilitate the retrieval of more semantically similar documents. The RSVD-LSI implementation can be used to select those *irrelevant* documents for perturbation and subsequent filtering from the returned document list. Since the number of relevant documents to a query, n_q (Eq. 5.2), is known in this collection, $n_q + 5$ is used as the window size for observing returned documents by RF-LSI and RSVD-LSI in all experiments.

5.5 Comparisons

The three LSI implementations (C-LSI, RF-LSI and RSVD-LSI) are compared via the average precision measure (see Section 5.3) and a precision-recall graph. Table 5.2 and 5.3 list all 30 available queries in MEDLINE. A comparison of the three implementations on individual queries is given by Table 5.4. For each query, the *highest* average precision is highlighted in boldface type.

Compared to the C-LSI model, the RSVD-LSI approach significantly improves the retrieval performance on almost all queries with the exception of query 1 (Table 5.2) where a surprisingly high average precision is achieved by C-LSI. This scenario demonstrates exactly when and where the RSVD-LSI approach might be appropriate for enhancing the retrieval effectiveness.

As also expected, the RSVD-LSI implementation successfully outperforms the RF-LSI model on most of the queries, and is on a par with RF-LSI for queries 1 (Table 5.2), 26 and 27 (Table 5.3). The only exceptions to this favorable comparison for RSVD-LSI are queries 10 (Table 5.2), 16, and 28 (Table 5.3). For these

Table 5.2: The 30 available queries in MEDLINE (query 1 - query 15).

	Query
1	the crystalline lens in vertebrates, including humans.
2	the relationship of blood and cerebrospinal fluid oxygen concentrations or partial pressures. a method of interest is polarography.
3	electron microscopy of lung or bronchi.
4	tissue culture of lung or bronchial neoplasms.
5	the crossing of fatty acids through the placental barrier. normal fatty acid levels in placenta and fetus.
6	ventricular septal defect occurring in association with aortic regurgitation.
7	radioisotopes in heart scanning. mainly used in diagnosis of pericardial effusions. also used to study tumors, heart enlargement, aneurysms and pericardial thickening. technetium, rihsa, radioactive.
8	the effects of drugs on the bone marrow of man and animals, specifically the effect of pesticides. also, the significance of bone marrow changes. hippurate, cholegraffin are used.
9	the use of induced hypothermia in heart surgery, neurosurgery, head injuries and infectious diseases.
10	neoplasm immunology.
11	blood or urinary steroids in human breast or prostatic neoplasms.
12	effect of azathioprine on systemic lupus erythematosus, particularly in regard to renal lesions.
13	bacillus subtilis phages and genetics, with particular reference to transduction.
14	renal amyloidosis as a complication of tuberculosis and the effects of steroids on this condition. only the terms kidney diseases and nephrotic syndrome were selected by the requester. prednisone and prednisolone are the only steroids of interest.
15	homonymous hemianopsia in visual aphasia, particularly measurement and assessment. gerstmann's syndrome and agnosia are also of interest.

Table 5.3: The 30 available queries in MEDLINE (query 16 - query 30).

	Query
16	separation anxiety in infancy (i.e. up to two years of age) and in preschool children, particularly separation of a child from its mother.
17	nickel in nutrition: requirements for methods for analysis; relation with enzyme systems; toxicity of, in humans and laboratory animals; deficiency signs and symptoms; level in various foodstuffs; level in blood and tissues.
18	the toxicity of organic selenium compounds.
19	excretion of phosphate or pyrophosphate in the urine or the effect of parathyroid hormone on kidney.
20	somatotropin as it effects bone, bone development, regeneration, resorption, bone cells, osteogenesis, physiologic calcification or ossification, cartilage and bone diseases in general. somatotropin as it relates to hypophysectomy, pituitary function, diseases, dwarfism, neoplasms, hypopituitarism and hyperpituitarism, and growth in general.
21	language development in infancy and pre-school age.
22	mycoplasma (infection or presence) in embryo, fetus, newborn infant or animal, or in pregnancy, gynecologic diseases, or as related to chromosomes or chromosome abnormalities.
23	infantile autism.
24	compensatory renal hypertrophy - stimulus resulting in mass increase (hypertrophy) and cell proliferation (hyperplasia) in the remaining kidney following unilateral nephrectomy in mammals.
25	use of chlorothiazide (diuril) or hydrochlorothiazide (hydrodiuril) in the treatment of nephogenic diabetes insipidus in children; also, use of low sodium diets and aldactone (spironolactone) in the treatment of childhood nephogenic diabetes insipidus.
26	methods for experimental production of and known causes of hydrocephalus in animals and humans.
27	interested in the parasitic diseases. filaria parasites found in primates, the insect vectors of filaria, the related diptera, i. e., culicoides, mosquitos, etc., that may serve as vectors of this infection-disease; also the life cycles and transmission of the filaria. parasites and ecology of the taiwan monkey, macaca cyclopis, with emphasis on the filarial parasite, macacanema formosana.
28	palliation (temporary improvement) of cancer patients by using drugs, x-ray, surgery.
29	hereditary implications of prolonged neonatal obstructive jaundice associated with liver pathology. there are but two relatively common well-defined pathologic entities that present as such: 1) bile duct or biliary atresia and 2) giant cell transformation of the liver often referred to as neonatal hepatitis. information regarding liver and bile duct embryogenesis will be an important adjunct to papers on the disease processes.
30	hemophilia and christmas disease, especially in regard to the specific complication of pseudotumor formation (occurrence, pathogenesis, treatment, prognosis).

Table 5.4: Average precision of individual queries for C-LSI, RF-LSI and RSVD-LSI implementations on the MEDLINE collection.

Query	Average Precision		
	C-LSI	RF-LSI	RSVD-LSI
1	.9363	**.9608**	.9342
2	.6856	.7890	**.8265**
3	.7184	.7582	**.8929**
4	.6856	.6610	**.8202**
5	.8278	.8642	**.9008**
6	.8671	.8822	**.9311**
7	.7252	.7640	**.8009**
8	.3660	.4409	**.7789**
9	.7450	.7905	**.9025**
10	.3910	**.7694**	.4722
11	.7321	.8670	**.9451**
12	.2724	.3280	**.3460**
13	.9352	.9477	**.9959**
14	.7559	.7816	**.9280**
15	.7603	.8010	**.8589**
16	.5580	**.7205**	.6097
17	.4538	.7163	**.7405**
18	.4464	.5315	**.6905**
19	.6159	.7140	**.7334**
20	.8122	.8916	**.9454**
21	.4662	.5238	**.5987**
22	.5886	.7205	**.8437**
23	.8701	.8260	**.9460**
24	.8635	.9090	**.9488**
25	.8447	.8956	**.9231**
26	.4724	**.7527**	.7456
27	.7118	**.8781**	.8704
28	.8610	**.9610**	.8674
29	.8127	.8259	**.9005**
30	.5587	.5199	**.7644**

Table 5.5: Mean average precision over queries for C-LSI, RF-LSI and RSVD-LSI implementations on the MEDLINE collection.

Method	Mean Average Precision
C-LSI	.6780
RF-LSI	.7597
RSVD-LSI	.8154

three queries, RF-LSI performs very well in retrieving relevant documents. The explanation for this RF-LSI behavior is that those returned relevant documents, which are used to construct a relevance feedback query, contain significantly important terms which help retrieve more semantically similar documents. For example, it has been observed that the initial query 10 (*neoplasm immunology*) was greatly enriched by supplementing several weighty terms such as *cancer, tumor, antigen, cell* and *growth* that occur in those returned relevant documents.

Using the data in Table 5.4, the mean average precision over 30 available queries for each implementation is provided in Table 5.5. The mean average precision provides an overall performance measure for an information retrieval method. Table 5.5 indicates that for the MEDLINE collection, the RSVD-LSI method has achieved more than 20% better performance than the C-LSI model, and more than 7% better performance than the RF-LSI approach, respectively.

As discussed in Section 5.3, recall and precision are the most commonly used measures in information retrieval. In order to compare retrieval performance in terms of both recall and precision, a precision-recall graph - bivariate plot where the one axis is recall and the other precision - can be used to evaluate them simultaneously. Such graphs can be constructed for individual queries and/or over all queries in a collection.

The precision-recall curves for the C-LSI, RF-LSI and RSVD-LSI methods are shown in Figure 5.5. Precision is plotted as a function of recall for nine levels of recall, from 0.10 through 0.90. These curves represent average performance over the 30 queries available with the MEDLINE collection (precision-recall curves on individual queries are shown in Appendix B). They are typical precision-recall curves which show that recall and precision are inversely related, i.e., when precision goes up, recall typically goes down and vice versa. For all levels of recall, precision of the RSVD-LSI method lies well above that obtained with the C-LSI model, and the gap between these two curves is especially significant at recall levels from 0.20 through 0.80. For all but the lowest recall level (0.10), the RSVD-LSI method demonstrates a higher precision than that received from the RF-LSI approach. There is a moderate gap between RSVD-LSI and RF-LSI curves. These results are very consistent with those obtained via an interpolated average precision.

The difference in performance between RSVD-LSI and other LSI implementations is especially impressive at recall levels from 0.30 through 0.70. The relatively modest performance of the RSVD-LSI implementation at the lowest and highest levels of recall can be traced to at least two factors. First, precision is quite high

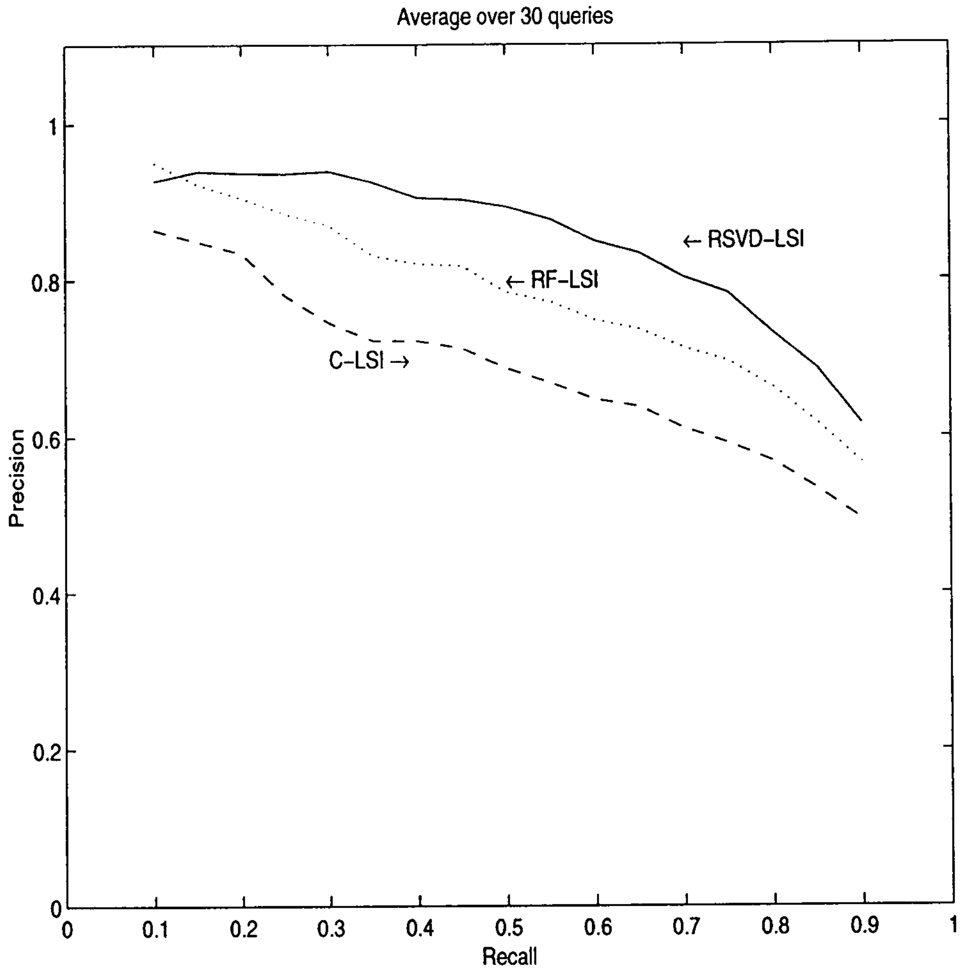


Figure 5.1: Precision-recall curves for C-LSI, RF-LSI and RSVD-LSI on the MEDLINE collection.

in all retrieval systems at low recall level, and, as an important strength, the original C-LSI (and RF-LSI) model has achieved a good precision at high recall level [DDF+90], leaving little room for improvement. Second, the filtering mechanism in RSVD-LSI is primarily designed to handle polysemy problems and these problems may not be as prevalent at low recall. Since only a relatively small number of irrelevant documents are selected and perturbed for each query, the precision improvement for RSVD-LSI at high recall is constrained.

The results on the MEDLINE collection can be summarized as follows: the RSVD-LSI is an updated LSI implementation equipped with an information filtering mechanism and greatly outperforms the C-LSI and RF-LSI methods. The original C-LSI is primarily designed to overcome synonymy problems (improving recall), while the nonlinear filtering mechanism based on R-SVD in RSVD-LSI (see Section 4.3) intends to handle polysemy problems (improving precision). As a fine combination of these two approaches, RSVD-LSI has been demonstrated as an effective and robust information retrieval/filtering technique.

Chapter 6

Conclusions and Future Work

A nonlinear generalization of the singular value decomposition known as the Riemannian SVD (R-SVD) [Moo93] has been discussed. It has numerous applications in noisy recognition, system identification, and parameter estimation.

The R-SVD is generalized to the low-rank matrices and used to formulate an enhanced LSI implementation for applications in information retrieval. LSI is an SVD-based conceptual retrieval technique which employs a rank-reduced model of the original (sparse) term-by-document matrix. Based on the appropriate term weighting schemes, LSI with relevance feedback (RF-LSI) could improve the retrieval performance quite significantly. However, the frequent reassignment of weights for large and rapidly changing document collections could also be computationally expensive. The proposed LSI implementation (RSVD-LSI) is capable of incorporating nonlinear perturbations to an existing rank-reduced model, and can effectively filter irrelevant documents from the top-ranked list of returned documents in the corresponding perturbed term-by-document space.

Experiments have shown that RSVD-LSI provides a highly effective information retrieval technique and demonstrates a new method of updating LSI (and perhaps similar vector-space information retrieval models) which could not be represented by traditional matrix decomposition tools such as the SVD. In experiments with a common benchmark test collection (MEDLINE), the RSVD-LSI implementation has achieved more than 20% better performance than the original LSI model, and more than 7% better performance than the RF-LSI approach for information retrieval, respectively.

Although considerable analysis and testing have been done with RSVD-LSI, an implementation of this new LSI model in a client-server software environment such as LSI++ [LB97] would be necessary to make it a commercially viable information retrieval system. Such an implementation could also provide a more accurate analysis in its retrieval performance and the directions for further improvement. Given that LSI++ is an efficient and flexible implementation for LSI, the addition of the nonlinear perturbation mechanism in RSVD-LSI to LSI++ would enhance the usability and retrieval effectiveness of the current LSI++ search engine.

It has been noted that some matrix-matrix and matrix-vector computations in Algorithm RSVD-LR2 can potentially be parallelized. A distributed implementation of RSVD-LSI on a parallel computer or a network of workstations is possible.

Determining the optimal dimension of the perturbed term-by-document space for effective filtering, and further analysis of Algorithm RSVD-LR2 for generating the *best* perturbation model constitute future research.

Bibliography

Bibliography

[BDO95] M. Berry, S. Dumais, and G. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review*, 37(4):573–595, 1995.

[Ber92] M. Berry. Large Scale Singular Value Decompositions. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[CCH92] J. Callan, B. Croft, and S. Harding. The INQUERY Retrieval System. *Proceedings of the Third International Conference on Database and Expert Systems Applications*, 78-83, Springer-Verlag, 1992.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(2):391–409, 1990.

[Dum91] Susan T. Dumais. Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments & Computers*, 23(2):229–236, 1991.

[EY36] C. Eckart and G. Young. The Approximation of One Matrix by Another Lower Rank. *Psychometrika*, 1:211–218, 1936.

[FBY92] W. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms.* Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[GL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[Har95] D. K. Harman. Overview of the Third Text Retrieval Conference (TREC-3). Technical Report NIST SP 500-225, National Institute of Standards and Technology, Gaithersburg, MD, April 1995.

[HPR96] S. Van Huffel, H. Park, and J. B. Rosen. Formulation and Solution of Structured Total Least Norm Problems for Parameter Estimation. *IEEE Trans. on Signal Processing*, 44(10):2464–2474, 1996.

[HV91] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problems: Computational Aspects and Analysis.* SIAM, Philadelphia, PA, 1991.

[Kow97] Gerald Kowalski. *Information Retrieval Systems:Theory and Implementation.* Kluwer Academic Publishers, Boston/Dordrecht/London, 1997.

[LB97] Todd A. Letsche and Michael W. Berry. Large Scale Information Retrieval with Latent Semantic Indexing. *Information Sciences - Applications*, 100:105–137, 1997.

[Mir60] L. Mirsky. Symmetric Gage Function and Unitarily Invariant Norms. *Q. J. Math*, 11:50–59, 1960.

[Moo93] B. De Moor. Structured Total Least Squares and L_2 Approximation Problems. *Linear Algebra and its Applications*, 188, 189:163–205, 1993.

[RPG96] J. B. Rosen, H. Park, and J. Glick. Total Least Norm Formulation and Solution for Structured Problems. *Siam J. Matrix Anal. Appl.*, 17(1):110–126, 1996.

[SB90] G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal American Society for Information Sciences*, 41(4):288–297, 1990.

[Zha91] H. Zha. The Restricted Singular Value Decomposition of Matrix Triplets. *SIAM J. Anal. Appl.*, 12(1):172–194, 1991.

Appendices

Appendix A

Convergence Analysis of Algorithm RSVD-LR1

Algorithm RSVD-LR1 converges only for some R-SVD problems (Eq. 3.3) corresponding to a low-rank matrix A_k, and some divergent cases have been identified.

Without loss of generality, assume that the perturbation matrix P in (Eq. 3.12) has a rectangular nonzero cluster which is located in the bottom-right corner of P and has the dimension of l by w (see Section 3.3.1). Then, the matrix D_v can be written as

$$\begin{aligned} D_v &= \operatorname{diag}\left(P \operatorname{diag}(v)\, v\right) \\ &= \begin{pmatrix} O_{(m-l)\times(m-l)} & O_{(m-l)\times l} \\ O_{l\times(m-l)} & \alpha\, I_{l\times l} \end{pmatrix}, \end{aligned} \tag{A.1}$$

where $O_{(m-l)\times(m-l)} \in \mathcal{R}^{(m-l)\times(m-l)}$, $O_{(m-l)\times l} \in \mathcal{R}^{(m-l)\times l}$ and $O_{l\times(m-l)} \in \mathcal{R}^{l\times(m-l)}$ are zero submatrices, and $I_{l\times l} \in \mathcal{R}^{l\times l}$ is an identity matrix and α is some constant.

Assume $l = m-k$ and let $U_{m-k} = (u_1, u_2, \ldots, u_m)^T \in \mathcal{R}^{m\times(m-k)}$ in (Eq.3.11), then

$$\begin{aligned} D^T_{m-k} D_v U_{m-k} &= \begin{pmatrix} u_{m-l+1} \\ u_{m-l+2} \\ \vdots \\ u_m \end{pmatrix} D_v \begin{pmatrix} u_{m-l+1} & u_{m-l+2} & \ldots & u_m \end{pmatrix} \\ &= \alpha (U^{(l)}_{m-k})^T U^{(l)}_{m-k}, \end{aligned} \tag{A.2}$$

where $U^{(l)}_{m-k}$ is constructed from U_{m-k} by truncating its first $m-l$ rows, and hence has the dimension of l by l. Since $U^{(l)}_{m-k}$ is a square matrix (assuming it is invertible),

$$(U^T_{m-k} D_v U_{m-k})^{-1} = \frac{1}{\alpha} (U^{(l)}_{m-k})^{-1} (U^{(l)}_{m-k})^{-T}.$$

Let U_{m-k} be partitioned into a two-block structure as

$$U_{m-k} = (U^{(m-l)}_{m-k}, U^{(l)}_{m-k})^T$$

(i.e., the first block contains first $m-l$ rows of U_{m-k} while the second contains the remaining l rows), then

$$U_{m-k}(U_{m-k}^T D_v U_{m-k})^{-1} U_{m-k}^T = \frac{1}{\alpha} \begin{pmatrix} U_{m-k}^{(m-l)} \\ U_{m-k}^{(l)} \end{pmatrix} (U_{m-k}^{(l)})^{-1} (U_{m-k}^{(l)})^{-T} \tag{A.3}$$

$$\begin{pmatrix} (U_{m-k}^{(m-l)})^T & (U_{m-k}^{(l)})^T \end{pmatrix}$$

$$= \frac{1}{\alpha} \left[\begin{pmatrix} U_{m-k}^{(m-l)} \\ U_{m-k}^{(l)} \end{pmatrix} (U_{m-k}^{(l)})^{-1} \right] \left[(U_{m-k}^{(l)})^{-T} \begin{pmatrix} (U_{m-k}^{(m-l)})^T & (U_{m-k}^{(l)})^T \end{pmatrix} \right]$$

$$= \frac{1}{\alpha} \begin{pmatrix} U_{m-k}^{(m-l)} (U_{m-k}^{(l)})^{-1} \\ I_{l\times l} \end{pmatrix} \begin{pmatrix} (U_{m-k}^{(l)})^{-T} (U_{m-k}^{(m-l)})^T & I_{l\times l} \end{pmatrix}$$

$$= \frac{1}{\alpha} \begin{pmatrix} U_{m-k}^{(m-l)} (U_{m-k}^{(l)})^{-1} (U_{m-k}^{(l)})^{-T} (U_{m-k}^{(m-l)})^T & U_{m-k}^{(m-l)} (U_{m-k}^{(l)})^{-1} \\ (U_{m-k}^{(m-l)} (U_{m-k}^{(l)})^{-1})^T & I_{l\times l} \end{pmatrix}.$$

Now from (Eq. A.1), it can be shown that

$$U_{m-k}(U_{m-k}^T D_v U_{m-k})^{-1} U_{m-k}^T D_v = \begin{pmatrix} O_{(m-l)\times(m-l)} & U_{m-k}^{(m-l)} (U_{m-k}^{(l)})^{-1} \\ O_{l\times(m-l)} & I_{l\times l} \end{pmatrix}. \tag{A.4}$$

Then, Algorithm RSVD-LR1 produces

$$\begin{aligned} u^{(new)} &= [U_k - U_{m-k}(U_{m-k}^T D_v U_{m-k})^{-1} (U_{m-k}^T D_v U_k)]x \qquad (A.5) \\ &= U_k x - [U_{m-k}(U_{m-k}^T D_v U_{m-k})^{-1} U_{m-k}^T D_v] U_k x. \end{aligned}$$

From the derived expressions in (Eq. A.4) and (Eq. A.5) it is easy to see that the bottom l components of the updated singular vector $u^{(new)}$ must be zero-valued. Following Algorithm RSVD-LR1, the updated singular vector $v^{(new)}$ and then the matrices $D_u^{(new)}$ and $D_v^{(new)}$ vanish and in consequence, the iteration process terminates abruptly.

Now, assume $w = n - k$, then a similar derivation can be applied to the matrix V_{n-k} and the formula

$$v^{(new)} = [V_k - V_{n-k}(V_{n-k}^T D_u V_{n-k})^{-1}(V_{n-k}^T D_u V_k)]\, s$$

of (Eq. 3.11), and it can be found that Algorithm RSVD-LR1 will diverge in this case as well.

Furthermore, when $m - k > l$ or $n - k > w$, it can be observed from the above discussion that Algorithm RSVD-LR1 will also fail to converge to a solution of the R-SVD problems (Eq. 3.3). Therefore, under the assumption that the perturbation matrix P (Eq. 3.4) has a rectangular nonzero block with the dimension of l by w, it can be concluded that Algorithm RSVD-LR1 will surely fail to generate the meaningful approximation solution to (Eq. 3.3) *when* the dimension of the null space of A_k^T (i.e., $m - k$) is greater than or equal to l, or *when* the dimension of the null space of A_k (i.e., $n - k$) is greater than or equal to w.

Appendix B

More Precision-Recall Curves on MEDLINE

This appendix contains precision-recall curves for the C-LSI, RF-LSI and RSVD-LSI models on individual 30 queries from the MEDLINE collection (see Chapter 5). For each query, the precision-recall curve for C-LSI, RF-LSI and RSVD-LSI is represented by a dashed line, a dotted line, and a solid line, respectively.

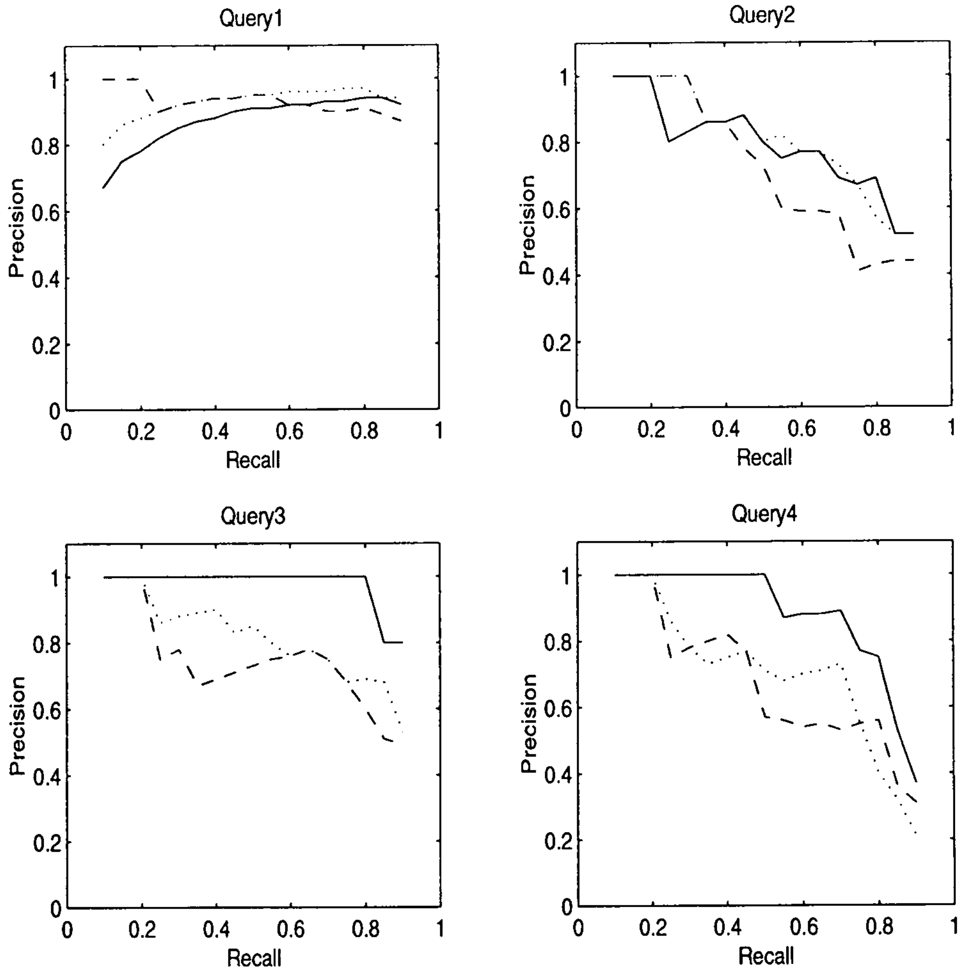


Figure B.1: Precision-recall curves for queries 1, 2, 3, 4 (**Query 1**: the crystalline lens in vertebrates, including humans; **Query 2**: the relationship of blood and cerebrospinal fluid oxygen concentrations or partial pressures. a method of interest is polarography; **Query 3**: electron microscopy of lung or bronchi; **Query 4**: tissue culture of lung or bronchial neoplasms).

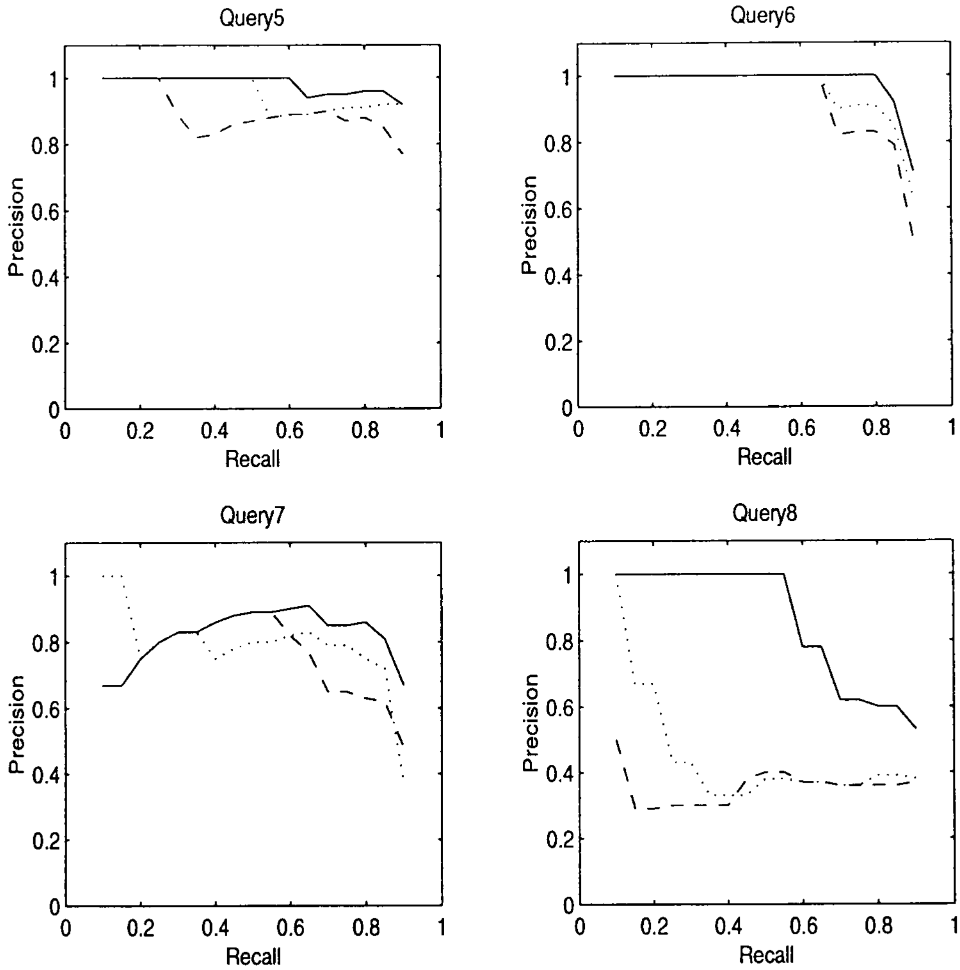


Figure B.2: Precision-recall curves for queries 5, 6, 7, 8 (**Query 5**: the crossing of fatty acids through the placental barrier. normal fatty acid levels in placenta and fetus; **Query 6**: ventricular septal defect occurring in association with aortic regurgitation; **Query 7**: radioisotopes in heart scanning. mainly used in diagnosis of pericardial effusions. also used to study tumors, heart enlargement, aneurysms and pericardial thickening. technetium, rihsa, radioactive hippurate, cholegraffin are used; **Query 8**: the effects of drugs on the bone marrow of man and animals, specifically the effect of pesticides. also, the significance of bone marrow changes).

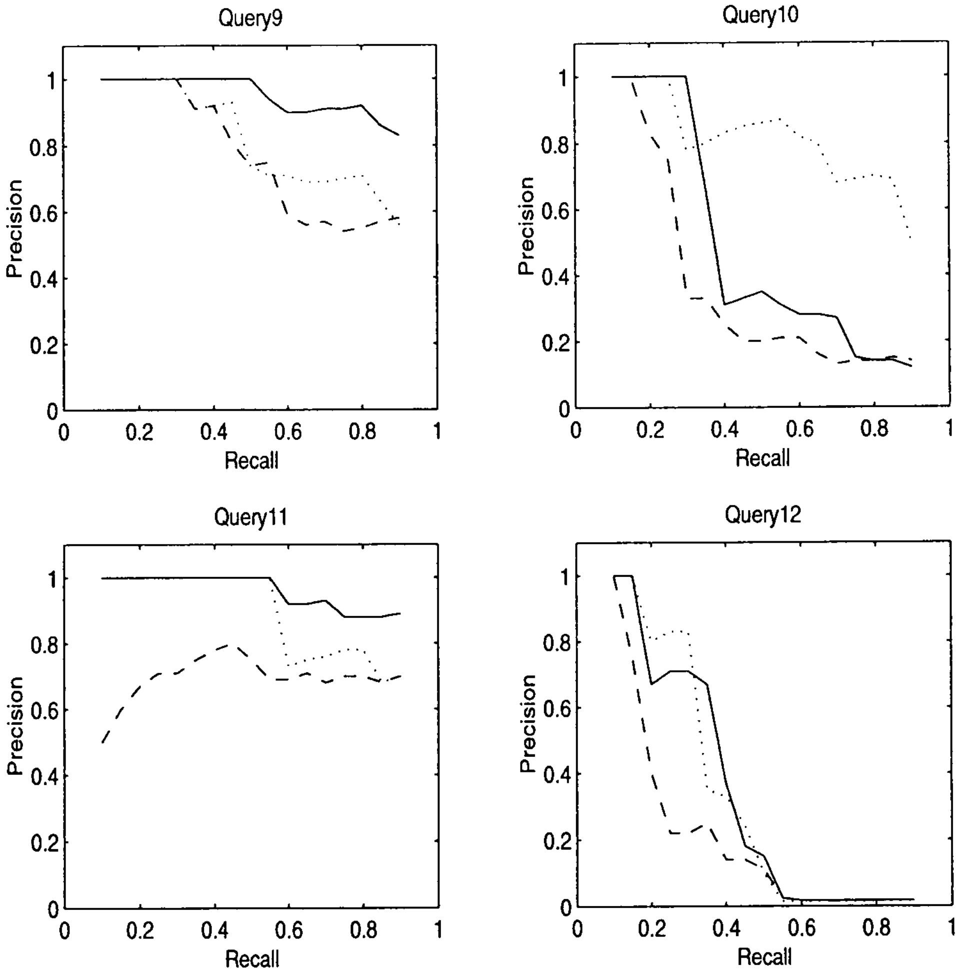


Figure B.3: Precision-recall curves for queries 9, 10, 11, 12 (**Query 9**: the use of induced hypothermia in heart surgery, neurosurgery, head injuries and infectious diseases; **Query 10**: neoplasm immunology; **Query 11**: blood or urinary steroids in human breast or prostatic neoplasms; **Query 12**: effect of azathioprine on systemic lupus erythematosus, particularly in regard to renal lesions).

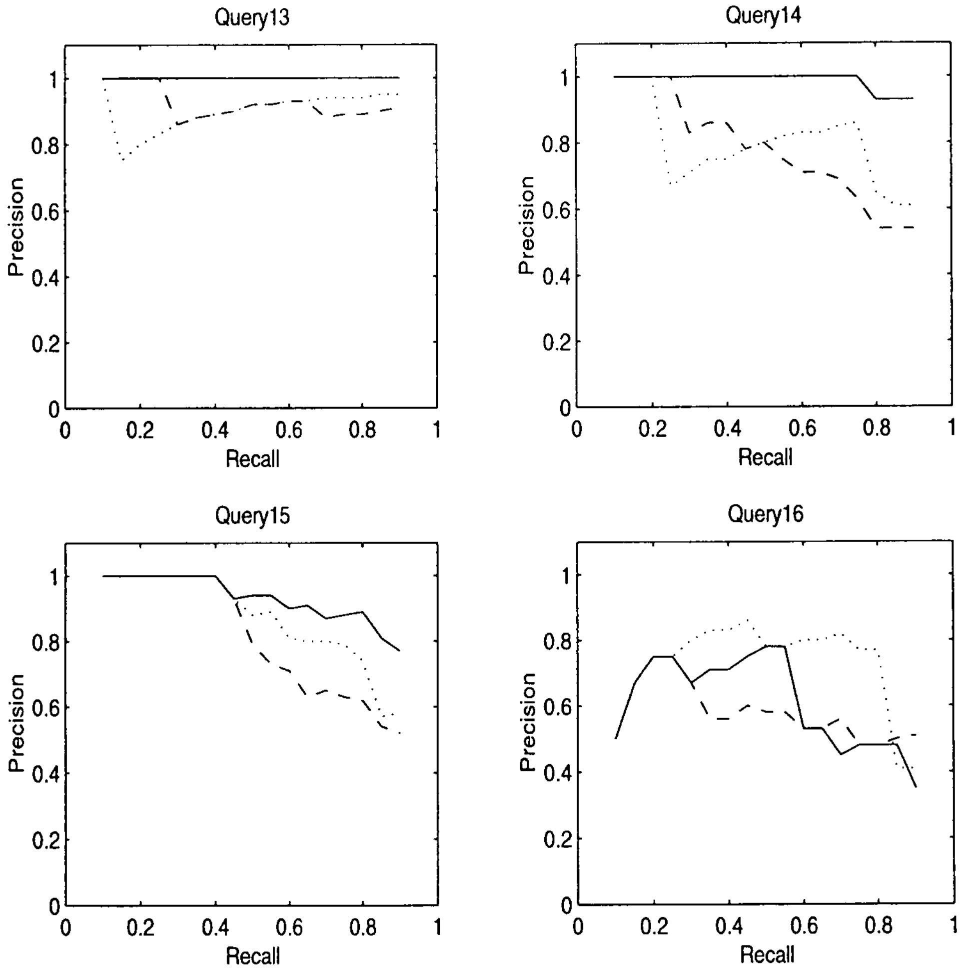


Figure B.4: Precision-recall curves for queries 13, 14, 15, 16 (**Query 13**: bacillus subtilis phages and genetics, with particular reference to transduction; **Query 14**: renal amyloidosis as a complication of tuberculosis and the effects of steroids on this condition. only the terms kidney diseases and nephrotic syndrome were selected by the requester. prednisone and prednisolone are the only steroids of interest; **Query 15**: homonymous hemianopsia in visual aphasia, particularly measurement and assessment. gerstmann's syndrome and agnosia are also of interest; **Query 16**: separation anxiety in infancy (i.e. up to two years of age) and in preschool children, particularly separation of a child from its mother).

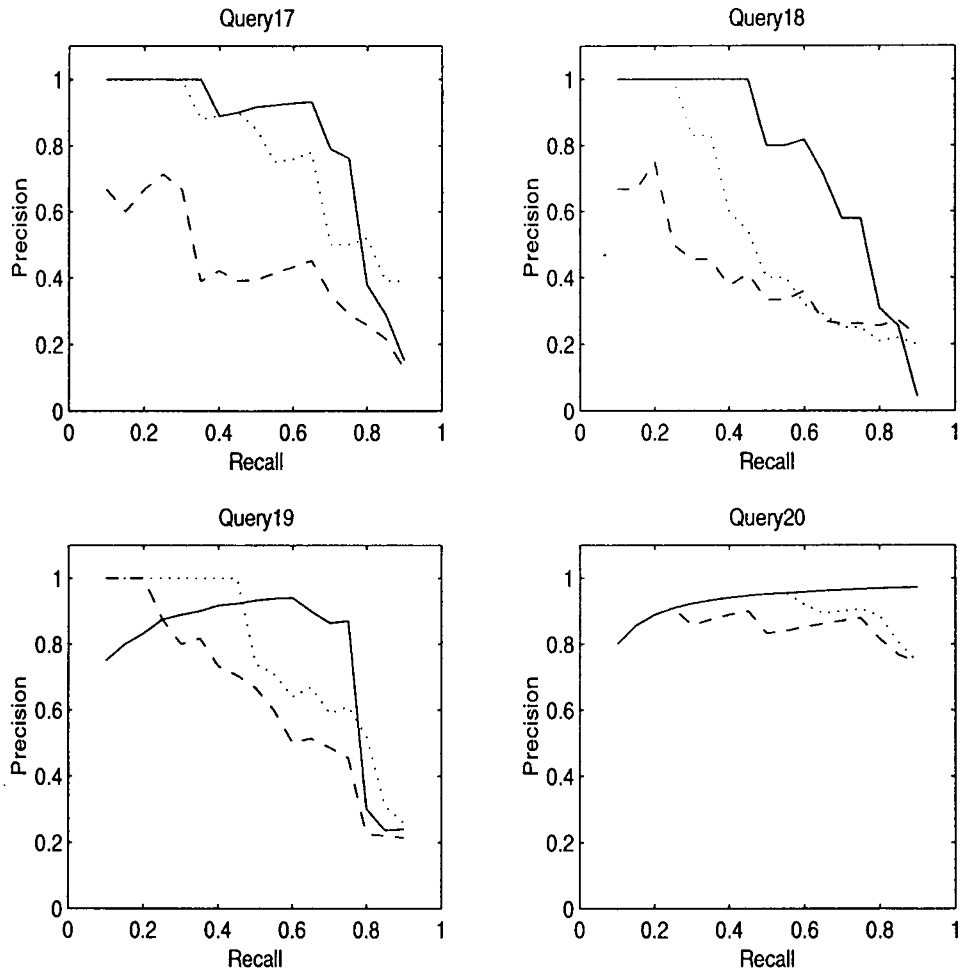


Figure B.5: Precision-recall curves for queries 17, 18, 19, 20 (**Query 17**: nickel in nutrition: requirements for methods for analysis; relation with enzyme systems; toxicity of, in humans and laboratory animals; deficiency signs and symptoms; level in various foodstuffs; level in lood and tissues; **Query 18**: the toxicity of organic selenium compounds; **Query 19**: excretion of phosphate or pyrophosphate in the urine or the effect of parathyroid hormone on kidney; **Query 20**: somatotropin as it effects bone, bone development, regeneration,resorption, bone cells, osteogenesis, physiologic calcification or ossification, cartilage and bone diseases in general. somatotropin as it relates to hypophysectomy, pituitary function, diseases, dwarfism,neoplasms, hypopituitarism and hyperpituitarism, and growth in general).

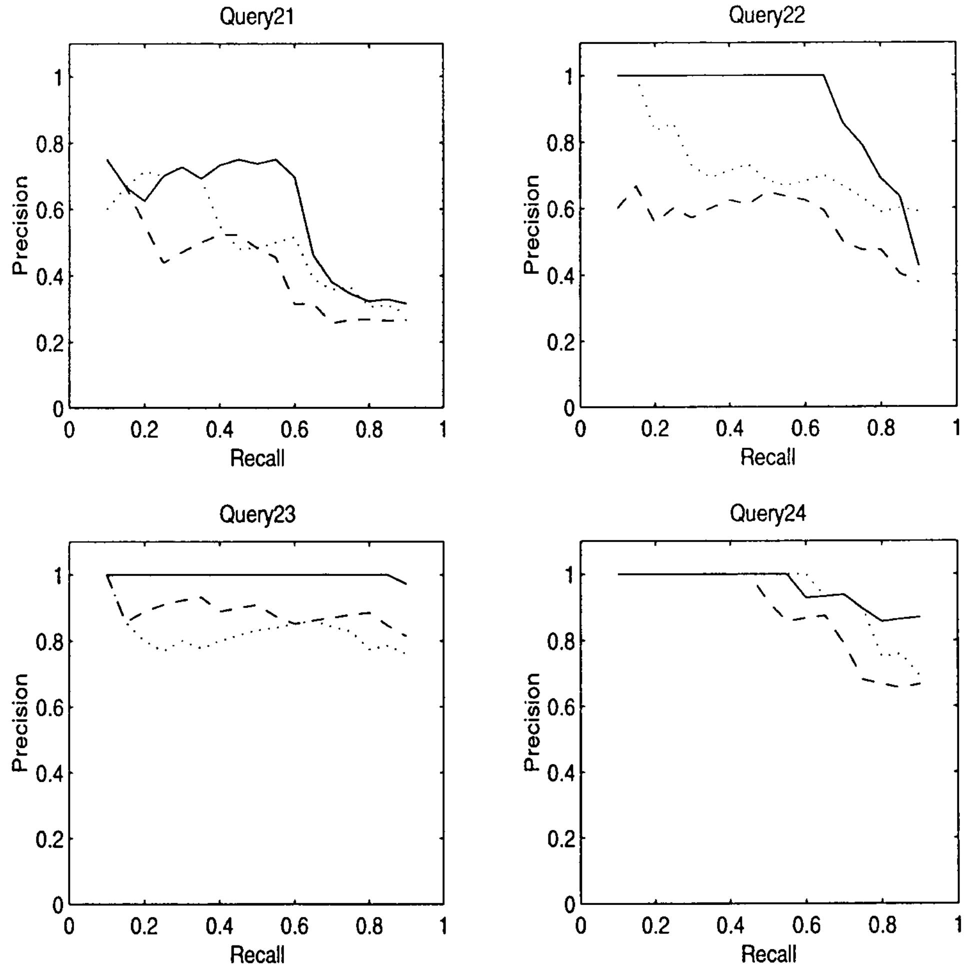


Figure B.6: Precision-recall curves for queries 21, 22, 23, 24 (**Query 21**: language development in infancy and pre-school age; **Query 22**: mycoplasma (infection or presence) in embryo, fetus, newborn infant or animal, or in pregnancy, gynecologic diseases, or as related to chromosomes or chromosome abnormalities; **Query 23**: infantile autism; **Query 24**: compensatory renal hypertrophy - - stimulus resulting in mass increase (hypertrophy) and cell proliferation (hyperplasia) in the remaining kidney following unilateral nephrectomy in mammals).

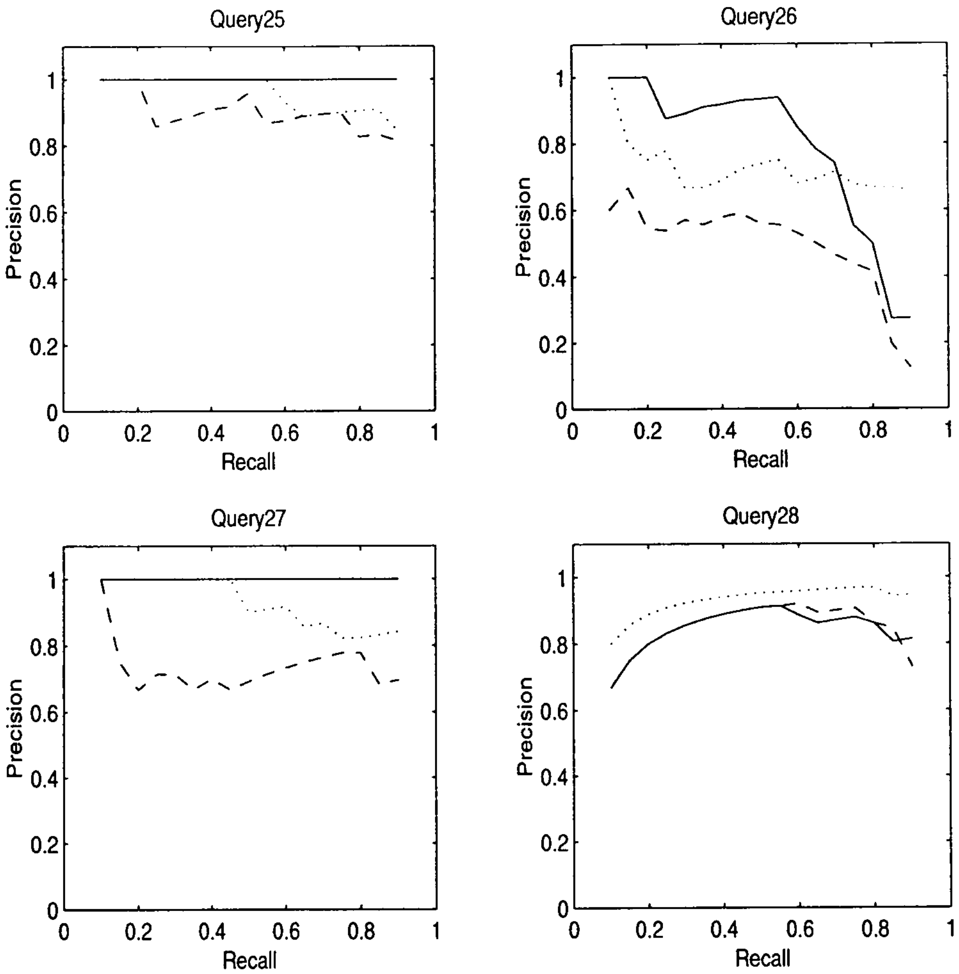
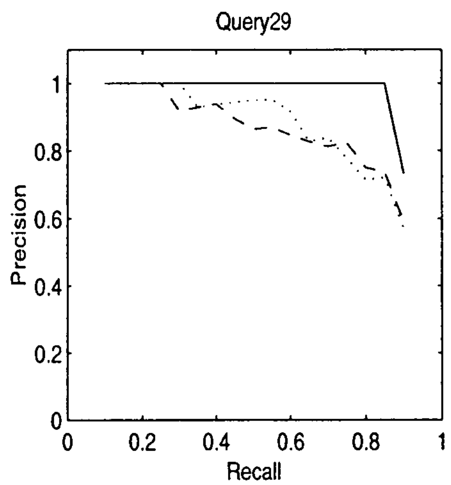


Figure B.7: Precision-recall curves for queries 25, 26, 27, 28 (**Query 25**: use of chlorothiazide (diuril) or hydrochlorothiazide (hydrodiuril) in the treatment of nephogenic diabetes insipidus in children; also, use of low sodium diets and aldactone (spironolactone) in the treatment of childhood nephogenic diabetes insipidus; **Query 26**: methods for experimental production of and known causes of hydrocephalus in animals and humans; **Query 27**: interested in the parasitic diseases. filaria parasites found in primates, the insect vectors of filaria, the related diptera, i.e., culicoides, mosquitos, etc., that may serve as vectors of this infection-disease; also the life cycles and transmission of the filaria. parasites and ecology of the taiwan monkey, macaca cyclopis, with emphasis on the filarial parasite, macacanema formosana; **Query 28**: palliation (temporary improvement) of cancer patients by using drugs, x-ray, surgery).



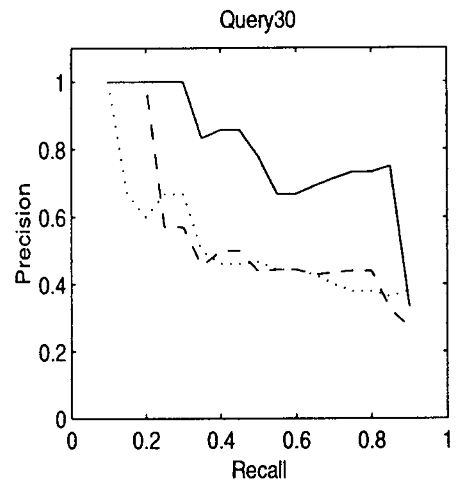


Figure B.8: Precision-recall curves for queries 29, 30 (**Query 29**: hereditary implications of prolonged neonatal obstructive jaundice associated with liver pathology. there are but two relatively common well-defined pathologic entities that present as such: 1) bile duct or biliary atresia and 2) giant cell transformation of the liver often referred to as neonatal hepatitis. information regarding liver and bile duct embryogenesis will be an important adjunct to papers on the disease processes ; **Query 30**: hemophilia and christmas disease, especially in regard to the specific complication of pseudotumor formation (occurrence, pathogenesis, treatment, prognosis).

Vita

Eric Peiqing Jiang was born in Shanghai, China on October 10, 1959. He graduated from Fen Yu High School in 1978, and received a Bachelor of Science degree in Applied Mathematics in 1982 from the Shanghai Chiao Tong University in Shanghai, China. He received a Master of Arts degree in Mathematics in 1989 and a Master of Science degree in Computer Science in 1992, respectively, from the University of Georgia in Athens, Georgia. In August 1998 he received his Doctor of Philosophy degree in Computer Science from the University of Tennessee in Knoxville, Tennessee.