CHALMERS | GÖTEBORGS UNIVERSITET

PREPRINT

Modified Krylov Subspace Methods for Information Retrieval

KATARINA BLOM

Department of Mathematical Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
GÖTEBORG UNIVERSITY
Göteborg Sweden 2004

Preprint 2004:21

Modified Krylov Subspace Methods for Information Retrieval

Katarina Blom

CHALMERS | GÖTEBORGS UNIVERSITET

Mathematics
Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
Göteborg, April 2004

NO 2004:21
ISSN 0347-2809

Matematiska Vetenskaper
Göteborg 2004

Modified Krylov subspace methods for information retrieval.

Katarina Blom

April 12, 2004

Abstract

This paper describes how simple modifications of the Krylov subspace method for IR can be used to steer what documents to retrieve and thus improve retrieval performance.

In our experiments retrieval performance, measured in average precision, is in general better for those queries that are at a smaller angle to their subspaces of relevant documents.

Improved query vectors are used directly in the vector model to rank documents for relevancy, and also for explicit restart of the bidiagonalization procedure in the Krylov subspace method.

The bidiagonalization process used in the Krylov subspace method is rewritten so that only directions orthogonal to subspaces spanned by irrelevant documents will be taken into account.

Starting the bidagonalization procedure with subspaces spanned by the terms in the queries, or a block of relevant retrieved documents will further improve the ranking of relevant documents. We replace the Golub-Kahan bidiagonalization procedure in the Krylov subspace method for IR with a band Lanczos procedure.

The modifications we make are based on relevance feedback and quite naturally our experiments show a significant increase in retrieval performance for the modified methods compared to the original Krylov subspace method used for IR.

Keywords
Information retrieval, Relevance feedback, Lanczos algorithm, band Lanczos algorithm, Vector space model, Krylov subspace, SVD, Singular value decomposition, query expansion, numerical linear algebra.

Contents

1 Introduction

An information retrieval (IR) system matches user queries (formal statements of information needs) to documents stored in a database. For each query entered the IR system will rank all the documents in relevance order. In *vector space models* queries and documents are encoded as vectors in m-dimensional space, where m is the number of unique terms in the collection. Documents are sorted for relevancy measuring the angles between each document vector and the query vector. The vector model has some major drawbacks. The terms used in the query vectors are often not the same as those by which the information searched has been indexed in the document vectors. In the vector model all document vectors that have no term in common with the query vector will be orthogonal to the query vector (and there by be ranked as irrelevant).

In Blom and Ruhe [7, 6] we discussed how a few iterations in the Golub-Kahan bidiagonalizing procedure could (at least in theory) overcome this problem and further improve the document ranking. In this paper we describe how simple modifications of the methods used in [6] can be used to steer what documents to retrieve and thus improve retrieval performance.

In IR *query expansion* is used to change the terms in the query vector in order to formulate a new improved query vector Harman [13], Ide [15], Rochio [17], Xu and Croft [20]. The original query is replaced with the expanded query and the documents are ranked again. The expected effect is that the new query is moved towards the relevant documents and away from irrelevant documents and there by improve the ranking.

In our experiments there is a relationship between the query vectors closeness in angles to their relevant subspaces (the subspaces spanned by column vectors corresponding to relevant documents) and retrieval performance in vector models. For query vectors close in angles to to their relevant subspaces, often retrieval performance (measured in average precision) is better than for queries further away from their relevant subspaces.

We discuss how simple projections of the original query vector q can be used to formulate improved query vectors. We use the retrieved relevant documents from the vector model to create new query vectors that (hopefully) are closer in angles to the relevant subspace.

For Krylov subspace methods *explicit restart* means replacing the starting vector q with an improved starting vector q^+ and restart the bidiagonalization procedure with this new vector. (In eigenvalue computations explicit restart

is often used to limit the sizes of the basis set). We use the retrieved relevant documents from the Krylov subspace method to construct new starting vectors for the explicit restart.

The two alternating steps of matrix-vector multiplications, A^Tq and Ap, in the bidiagonalization procedure in the Krylov method can be roughly interpreted as finding all documents containing the terms in the query q and finding all the terms contained in the documents represented by p. It is easy to see that after several iterations this process will bring in some relevant documents as well as many irrelevant ones. Controlling this growth process will improve retrieval efficiency for the Krylov subspace method.

Technically it is easy to rewrite the bidiagonalizing procedure to exclude unwanted (irrelevant) search directions. We use the retrieved irrelevant documents to construct subspaces spanned by unwanted directions to avoid in the bidiagonalizing procedure.

Starting the bidiagonalizing procedure with a block of relevant documents (relevant directions) instead of only one vector will improve performance significantly. This is done by replacing the Golub-Kahan bidiagonalizing procedure used in the Krylov subspace method for IR with a band Lanczos algorithm.

This article is organized as follows:

In section 1.1 we present measures for retrieval efficiency used in this article. The measures are standard in the IR community. The notation and some symbols frequently used throughout the article are listed in section 1.2. In section 1.3 we give a short mathematical background for some concepts further used.

Section 2 gives a short presentation of the Krylov method used for IR [6]. (The Golub-Kahan bidagonalization procedure used in the Krylov method is shortly presented in appendix A).

In section 3 we introduce the *relevant subspace* (the subspace spanned by relevant document vectors) and it's complement. We discuss how the query vectors may be projected onto the relevant subspace or orthogonal to the complement. We also introduce an *optimal scoring*.

In section 4 we introduce techniques for how to approximate the projected query vectors from section 3 and how to approximate the optimal scoring.

In section 4.2 improved query vectors are used for explicit restart of the bidiagonalization procedure. We also discuss how the bidiagonalization procedure can be modified to search only in directions orthogonal to unwanted directions.

Section 4.3 introduce *query subspaces* (subspaces spanned by the terms in the queries). A short description of the band Lanczos procedure and how it can be used for IR is given.

In section 5 some numerical experiments are presented.

1.1 Measures

The retrieval efficiency of an information retrieval system depends on two main factors. The ability of the system to retrieve relevant information and the ability to dismiss irrelevant information. The ability to retrieve relevant information is measured by *recall*, the ratio of relevant documents retrieved over the total number of relevant documents for that query. A systems ability to reject irrelevant documents is measured by *precision*, the ratio of the number of relevant documents retrieved for a given query over the total number of documents retrieved. Precision and recall are usually inversely related (when precision goes up, recall goes down and vice versa).

When we evaluate a query q, all the documents are ranked and we receive an ordered list $\mathcal{L}$ of documents. Assume t documents are relevant to the query and let $\ell_i, \quad i = 1 \dots t$ be the position for the ith relevant document in $\mathcal{L}$. The *average precision* (non interpolated) for a single query is defined as

$$\frac{1}{t} \sum_{i=1}^{t} \frac{i}{\ell_i}.$$

The *mean average precision* for multiple queries is defined as the mean of the average precisions for all queries.

Precision can be computed at any *actual recall level*

$$\frac{i}{t}, \quad i = 1, 2, \dots, t$$

(where t is the number of relevant documents to the query).

Let r_j be the jth recall level from the 11 *standard recall levels* $0, 0.1, 0.2 \dots 1$. The *interpolated average precision* for a query at standard recall level r_j is the maximum precision obtained for any actual recall level greater that or equal to r_j.

The *Recall level precision averages* for multiple queries are the means of the interpolated average precision values at each (standard) recall level for

the queries. Recall level precision averages are used as input for plotting the recall-precision graphs.

For further details, see Harman [14].

1.2 Notation

The notation used in this article is rather standard in the numerical linear algebra community. We use uppercase letters for matrices and lowercase letters for vectors. Lowercase Greek letters usually denotes scalars. Component indices are denoted by subscript. For example, a vector c and a matrix M might have entries c_i and m_{ij} respectively. On the occasions when both an iteration index and a component index are needed, the iteration is indicated by a parenthesised superscript, as in $c_j^{(r)}$ to indicate the jth component of the rth vector in a sequence. Otherwise c_j may denote either the jth component of a vector c or the jth column of a matrix C. The particular meaning will be clear from its context.

The range of a matrix M is the subspace spanned by the columns of M and is denoted $R(M)$.

Some symbols are frequently used throughout the article. They are listed below. For a detailed description please see each reference.

A**:** Term document matrix (section 2).

q**:** Query vector (section 2).

$\mathcal{Q}$**:** The query subspace (section 4.3.1).

$\mathcal{A}$**:** We use $\mathcal{A}$ for $R(A)$, the range of A. It is spanned by the columns of A and has the dimension r, the rank of A.

$\mathcal{R}$**,** $\mathcal{C}$**:** The relevant subspace and the complementary subspace (section 3).

$\tilde{\mathcal{R}}$**,** $\tilde{\mathcal{C}}$**:** The subspace spanned by the relevant retrieved documents and its complement in the residual collection (section 4.1).

Some of the notations not listed here that are used throughout the article are introduced in section 1.3.

1.3 Mathematical background

Orthogonal projections Let M be any $m \times n$ matrix. The *orthogonal projection* of the column vectors in M onto the space $\mathcal{S}$ is denoted $\mathcal{P}_S M$. The column vectors in M may also be projected orthogonal to $\mathcal{S}$, $M - \mathcal{P}_S M$.

Principal angles The relative orientation of two subspaces can be described by *principal angles*, the angles formed by *principal vectors* in the spaces. Let $\mathcal{F}, \mathcal{G} \in R^m$ be two subspaces whos dimensions satisfy $p = dim(\mathcal{F}) \geq dim(\mathcal{G}) = s \geq 1$. The first principal angle θ_1 between $\mathcal{F}$ and $\mathcal{G}$ is the smallest angle that can be formed by a vector $f_1 \in \mathcal{F}$ and a vector $g_1 \in \mathcal{G}$. Since the angle is minimized when the cosine is maximized the smallest angle satisfies

$$\begin{aligned} \cos\theta_1 &= \max_{f\in\mathcal{F}} \max_{g\in\mathcal{G}} f^T g = f_1^T g_1 \\ \text{subject to} &\quad \|f\| = \|g\| = 1. \end{aligned}$$

The vectors f_1 and g_1 are called principal vectors. The second principal angle θ_2 is defined to be the smallest angle that can be formed between a vector in $\mathcal{F}$ that is orthogonal to f_1 and a vector in $\mathcal{G}$ that is orthogonal to g_1. The principal angles $\theta_1, \ldots, \theta_s$ between $\mathcal{F}$ and $\mathcal{G}$ are defined recursively by

$$\begin{aligned} \cos\theta_k &= \max_{f\in\mathcal{F}} \max_{g\in\mathcal{G}} f^T g = f_k^T g_k \\ \text{subject to} & \\ \|f\| &= \|g\| = 1, \\ f^T f_i &= 0, \ i = 1 \ldots k-1, \\ g^T g_i &= 0, \ i = 1 \ldots k-1. \end{aligned}$$

For further reading on principal angles see Watkins [19].

Let F be an $m \times p$ matrix and let G be an $m \times s$ matrix. Assume that both F and G have linearly independent columns and let $F = Q_F R_F$ and $G = Q_G R_G$ be the QR-decompositions of F and G respectively. Using the SVD the principal angles and vectors for the ranges $R(F)$ and $R(G)$ can be computed [4]. Let

$$USV^T = Q_F^T Q_G \tag{1}$$

be the singular value decomposition of $Q_F^T Q_G$. The cosines of the principal angles are the singular values in S and the principal vectors for F and G are $Q_F U$ and $Q_G V$ respectively.

Krylov subspaces The *Krylov subspace* $\mathcal{K}_r(B, x)$ of the square matrix B and starting vector x is spanned by the r vectors

$$x, Bx, B^2x, \ldots, B^{r-1}x$$

where x is any nonzero starting vector. The *block Krylov subspace* $\mathcal{K}_r(B, X)$ is spanned by the pr vectors in the block Krylov sequence

$$X, BX, B^2X, \ldots, B^{r-1}X$$

where the columns in $X = \begin{bmatrix} x_1 & x_2 & \ldots & x_p \end{bmatrix}$ are linearly independent.

Theorem Let vectors $x_1, x_2, \ldots x_p$ be orthonormal and starting vectors for the p sequences spanning the Krylov subspaces $\mathcal{K}_r(B, x_1)$, $\mathcal{K}_r(B, x_2)$, ..., $\mathcal{K}_r(B, x_p)$ respectively.

Then the block Krylov subspace is the union of the p Krylov subspaces,

$$\mathcal{K}_r(B, X) = \bigcup_{j=1}^{p} \mathcal{K}_r(B, x_j)$$

proof The proof follows from a simple permutation of the vectors that span the block Krylov subspace, we have

$$\begin{aligned} \mathcal{K}_r(B, X) &= \text{span}\{x_1, x_2, \ldots, x_p, Bx_1, Bx_2, \ldots, Bx_p, \ldots, B^{r-1}x_1, B^{r-1}x_2, \ldots, B^{r-1}x_p\} \\ &= \text{span}\{x_1, Bx_1, \ldots, B^{r-1}x_1, x_2, Bx_2, \ldots, B^{r-1}x_2, \ldots, x_p, Bx_p, \ldots, B^{r-1}x_p\} \\ &= \bigcup_{j=1}^{p} \mathcal{K}_r(B, x_j). \end{aligned}$$

2 Krylov subspace methods for information retrieval

In vector space models both queries and documents are encoded as vectors in m-dimensional space, where m is the number of unique terms in the collection. The document vectors are stored as columns in an $m \times n$ term document matrix A. The query vectors are stored as $m \times 1$ vectors q and query matching can be viewed as a search in the column space of the term document matrix A.

In the *vector model* the documents are scored measuring the cosine of the angles between the query vector q and each document vector a_j in A,

$$c_j = \frac{q^T a_j}{\|q\|_2 \|a_j\|_2}, \quad j = 1, 2, \ldots, n. \tag{2}$$

The smaller the angle (i.e. the larger cosine value) the higher relevance score.

For the *Krylov subspace methods* we will use the Golub Kahan bidiagonalization procedure [12] applied to the term document matrix A starting at the query vector q to compute the two *basis matrices* Q_{r+1} and P_r and the $(r+1) \times r$ lower bidiagonal matrix $B_{r+1,r}$ satisfying

$$B_{r+1,r} = Q_{r+1}^T A P_r \tag{3}$$

The column vectors in the basis matrices Q_{r+1} and P_r span bases for the two Krylov subspaces $\mathcal{K}_{r+1}(AA^T, q)$, in the document space (spanned by the query q and the columns of A) and $\mathcal{K}_r(A^T A, A^T q)$, in the term space (spanned by the rows of A) respectively. The Golub-Kahan bidiagonalization procedure is further described in appendix A.

We let W form an orthonormal basis for the column vectors in the *reached subspace* $\text{span}(AP_r)$.

The reached subspace W, the basis matrices Q_{r+1}, P_r and the $B_{r+1,r}$ matrix are used to score the documents in relevance order to the query (see Blom Ruhe [6]). The similarity measures we use for this article are:

- In the *subspace projection measure* the documents in A are sorted according to their closeness measured in angles to the Krylov subspace $\mathcal{K}_{r+1}(AA^T, q)$. The relevance is measured using

$$c_j = \|Q_{r+1}^T a_j\|_2, \quad j = 1, 2, \ldots, n. \tag{4}$$

- A projected query vector

$$\hat{q} = WW^T q \tag{5}$$

is constructed using the reached subspace. In the *expanded query measure* the documents are scored using the angles between the expanded query vector $\hat{q}$ and each document vector in A,

$$c_j = \frac{\hat{q}^T a_j}{\|\hat{q}\|_2 \|a_j\|_2}, \quad j = 1, 2, \ldots, n. \tag{6}$$

Note that if the starting vector $q \in \mathcal{A}$, the range of A, then the projected query $\hat{q} = q$ and the cosines (6) are simply the vector model scoring (2).

3 The subspace spanned by relevant document vectors

When a user issues a search for information on a topic, the information retrieval system will start to return documents that are relevant from the system's point of view. From the user's perspective the total database will be divided logically into four parts. There will be relevant and irrelevant documents retrieved. And among the documents not retrieved there will be both relevant and irrelevant documents.

Let $\mathcal{A}$ be the subspace spanned by the column vectors in the term document matrix A and assume r documents are relevant to query q. The subspace

$$\mathcal{R} \subseteq \mathcal{A} \tag{7}$$

spanned by the document vectors in A that correspond to the r relevant documents we call the *relevant subspace*. We define the *complementary subspace* by the set difference between the range of A and the relevant subspace,

$$\mathcal{C} = \mathcal{A} - \mathcal{R}. \tag{8}$$

It is important to note that with this definition of the relevant subspace the relevant document vectors are completely in $\mathcal{R}$. The irrelevant document vectors are spanned by vectors both in the relevant subspace and in the complementary subspace.

The relevant subspace $\mathcal{R}$ for a query q is *discernible* if no irrelevant document vectors are completely in $\mathcal{R}$. For all sets we have studied[1] the relevant subspaces for all queries are discernible[2].

[1] The Adi, Cici, Cranfield and Medline data sets [9] (see also [2]) and the Financial Times and Congressional Records from the TREC data sets [1].

[2] How the document vectors in the term document matrices were set up is described in section 5.

3.1 Query vector and the relevant subspace

The query vector q can be divided into three orthogonal parts q_1, q_2 and q_3 where q_1 is in the relevant subspace (7) of q, q_2 is in $\mathcal{A}$, the range of A but not in the relevant subspace and q_3 is orthogonal to $\mathcal{A}$.

$$q = q_1 + q_2 + q_3 \tag{9}$$

and $\|q\|_2^2 = \|q_1\|_2^2 + \|q_2\|_2^2 + \|q_3\|_2^2$. In our experiments $\|q_3\|_2$ is in general larger than $\|q_1\|_2$ and $\|q_2\|_2$, and $\|q_2\|_2$ is in general larger than $\|q_1\|_2$.

It is not always the case that the query vector is closer to the subspace of relevant documents than the irrelevant document vectors are. Quite often we will find irrelevant document vectors making a smaller angle to the relevant subspace than the query vector itself.

This is clearly seen in figure 1 where cosines of the angles between the relevant subspace $\mathcal{R}$ and each document vector and the query vector are plotted. 15 irrelevant document vectors make slightly smaller angles to the relevant subspace than the query vector.

We have also found several query vectors being orthogonal to their relevant subspaces.

There is a tendency that average precision (using any of the ranking algorithms presented in this article) is better for queries (or expanded queries) close to their relevant subspaces (7) than for queries further away from their relevant subspaces.

When sorting the retrieval performance (measured in average precision for the vector model (2)) for each query vector q within a data set according to $\|q_1\|_2$ there is a relationship (see figure 2 (right plot))[3]. Average precisions for query vectors orthogonal to or with a small part in their relevant subspaces is very moderate. Average precisions for query vectors closer in angles to their relevant subspaces tend to be higher.

3.2 Projected query vectors

We can move the query vector q away from irrelevant documents (measured in angles) by projecting it orthogonal to the complementary subspace $\mathcal{C}$ (8)

$$q - \mathcal{P}_{\mathcal{C}} q = q_1 + q_3 \tag{10}$$

[3]Similar relationships seems to occur when sorting the retrieval performance for each query q according to $\frac{\|q_1\|_2}{\|q_2\|_2}$.

where q_1 and q_3 are the parts of q that are in the relevant subspace for q and orthogonal to the range of A respectively as defined in (9). We can move the query vector towards the relevant subspace by projecting it onto the relevant subspace $\mathcal{R}$ (7)

$$\mathcal{P}_{\mathcal{R}} q = q_1. \tag{11}$$

Clearly both the projected queries (10) (11) are orthogonal to the complementary subspace $\mathcal{C}$. Unless q is completely in the range of A the projected query vectors (10) (11) are not equal[4].

3.3 Optimal scoring

An *optimal scoring* will rank all relevant documents better than irrelevant documents.

Assume the relevant documents for a query span a discernible subspace. Since $\mathcal{P}_{\mathcal{R}} a_j = a_j$ for all relevant document vectors a_j, then optimality is obtained if all documents are sorted in descending order according to their angles to the relevant subspace

$$\frac{\|\mathcal{P}_{\mathcal{R}} a_j\|_2}{\|a_j\|_2} = \frac{\|a_j - \mathcal{P}_{\mathcal{C}} a_j\|_2}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n. \tag{12}$$

The angles between the projected query vector $\mathcal{P}_{\mathcal{R}} q$ and each document vector a_j in the term document matrix

$$\frac{(\mathcal{P}_{\mathcal{R}} q)^T a_j}{\|\mathcal{P}_{\mathcal{R}} q\|_2 \|a_j\|_2} = \frac{(\mathcal{P}_{\mathcal{R}} a_j)^T q}{\|\mathcal{P}_{\mathcal{R}} q\|_2 \|a_j\|_2} = \frac{(a_j - \mathcal{P}_{\mathcal{C}} a_j)^T q}{\|\mathcal{P}_{\mathcal{R}} q\|_2 \|a_j\|_2}, \quad j = 1, 2, \ldots, n \tag{13}$$

does not necessarily give an optimal scoring.

This is easily verified. For relevant document vectors a_j, the angle between the projected document $\mathcal{P}_{\mathcal{R}} a_j$ and the query vector q is the same as the angle between the document vector itself and the query vector, [5]

$$\frac{(\mathcal{P}_{\mathcal{R}} a_j)^T q}{\|a_j\|_2} = \frac{a_j^T q}{\|a_j\|_2}, \quad j = 1, 2, \ldots r.$$

[4]For the data sets we have studied, (the Adi, Cici, Cranfield and Medline data sets [9], the FT and the CR sets from TREC [1]), q is never completely in $\mathcal{A}$.

[5]Since $\|\mathcal{P}_{\mathcal{R}} q\|_2$ in the denominator is constant for all $j = 1, 2, \ldots, n$ it will not affect the scoring and can be omitted.

We cannot gurantee that this quantity, $\frac{(\mathcal{P}_{\mathcal{R}} a_j)^T q}{\|a_j\|_2}$, is smaller than the angles between the projected irrelevant document vectors and the query vector. Some of the irrelevant documents may be ranked higher than relevant documents.

Clearly the scoring (13) is not optimal. In experiments this scoring (not surprisingly) performs very well measured in average precision.

However, the documents forming the relevant subspace for a query are not known in advance (in fact we are trying to find them). In reality we can only compute approximations to the projected query vectors (10), (11) and rankings (12), (13).

4 Using relevance feedback

Both the vector model and the Krylov subspace method have a limited recall. Usually some relevant documents are retrieved to a query, but almost never all the relevant documents. In this section we will discuss some techniques that may be used together with the Krylov method to further increase the recall (mostly the techniques will also increase precision).

To steer the process of what documents to retrieve, we will use relevance feedback. In a relevance feedback cycle, the user is presented a list of retrieved documents, and after examining them, marks those that are relevant. The main idea is to use the information provided by the user to make a new (hopefully) improved search[6].

4.1 Formulating improved query vectors

Assuming that relevant documents resemble each other it is natural to formulate an initial query and to incrementally compute vectors of the relevant and/or the complementary subspaces.

Assume that t retrieved documents have been returned back to the user at some point. Assume that s of these were identified as relevant and $t-s$ were irrelevant. Let the columns of A_t correspond to the t retrieved documents.

[6]Relevance feedback can also be performed without involving a user. In *pseudo relevance feedback* new queries are constructed using the top retrieved documents, see for example Xu and Croft [20].

Let

$$\tilde{\mathcal{R}} \subseteq R(A_t) \tag{14}$$

be the subspace spanning the s retrieved relevant document vectors and let

$$\tilde{\mathcal{C}} = R(A_t) - \tilde{\mathcal{R}} \tag{15}$$

be it's complement in the range of A_t.

Clearly the subspace spanned by the relevant retrieved document vectors is in the relevant subspace (7), $\tilde{\mathcal{R}} \subseteq \mathcal{R}$. The complement $\tilde{\mathcal{C}}$ to $\tilde{\mathcal{R}}$ in $R(A_t)$ may not be completely in the complementary subspace $\mathcal{C}$ (8)

We will mimic the two projected query vectors (10) and (11) with the two *expanded query vectors*

$$q^- = q - \mathcal{P}_{\tilde{\mathcal{C}}} q \quad \text{and} \quad q^+ = \mathcal{P}_{\tilde{\mathcal{R}}} q \tag{16}$$

respectively. Note that q^+ is in $\mathcal{A}$, the range of A, while q^- might not be completely in $\mathcal{A}$. It is possible to add tuning constants to the expanded query vector q^-. Letting

$$q^- = \alpha q + \beta \mathcal{P}_{\tilde{\mathcal{C}}} q \tag{17}$$

and with appropriate choices of α and β retrieval performance for the expanded query vector q^- may improve.

If no relevant documents were found among the t best scored documents then it is not possible to form the expanded query vector q^+. One option could then be to look further down the ranked list of retrieved documents in order to find some relevant documents. Often this is wasted effort, the vector model will pull in too many irrelevant documents and there is an upper limit to how many documents we can expect a human user to judge for relevance. Forming the other expanded query vector q^- is also likely to fail since the query vector q will be projected orthogonal to the subspace $\tilde{\mathcal{C}}$. The number of retrieved documents t is then much less than the number of document vectors n in A and the subspace $\tilde{\mathcal{C}}$ will be a very poor approximation of the complementary subspace $\mathcal{C}$ (8).

There are three classic (and rather similar) ways to calculate an improved query vector for vector models [10].

In *standard Rochio* [17] the new query vector is computed using the original query vector q and the retrieved relevant document vectors in A_s and the retrieved irrelevant document vectors in A_{t-s}

$$q_{\text{ROCHIO}} = \alpha q + \frac{\beta}{s} A_s e_s - \frac{\gamma}{t} A_{t-s} e_{t-s}$$

where e_s and e_{t-s} are the vectors of ones and α, β and γ are used as tuning constants. Sometimes γ is set to 0. For the two other methods, *Ide Regular* and *Ide dec hi* [15] other tuning constants are used and for the Ide dec hi method the highest ranked irrelevant document vector is subtracted instead of the sum of the irrelevant document vectors.

The last vector $A_{t-s}e_{t-s}$ in the standard Rochio expansion can be divided in two parts, one part that is in the subspace spanned by the relevant retrieved documents $\tilde{\mathcal{R}}$ (14) and one part that is in the complement $\tilde{\mathcal{C}}$ (15).

The standard Rochio expansion can be formulated

$$q_{\text{ROCHIO}} = \alpha q + \mathcal{P}_{\tilde{\mathcal{R}}}(\frac{\beta}{s}A_s e_s - \frac{\gamma}{t}A_{t-s}e_{t-s}) - \frac{\gamma}{t}\mathcal{P}_{\tilde{\mathcal{C}}}A_{t-s}e_{t-s}. \tag{18}$$

Let the *residual collection*

$$A_{n-t} \tag{19}$$

be the columns of A with the t document vectors that correspond to the retrieved documents removed.

We cannot formally prove that any of the three expanded query vectors (16) – (18) are closer in angles to their relevant subspaces in the residual collection than the query vectors q. In our experiments[7] some trends are clear though. In general q^+ and q_{ROCHIO} are moved towards their relevant subspaces in the residual collection. For $\alpha = \beta = 1$ the expanded query vectors q^- are moved away from the irrelevant document vectors but rarely towards their relevant subspaces in the residual collection. If an expanded query vector is moved towards its relevant subspace in the residual collection retrieval performance, using any of the ranking algorithms presented in this article, is in general better than performance for the original query vector q.

In figure 1 we used the ten best scored documents from the vector model (2) and constructed the expanded query vector q^+ (16). The expanded query vector q^+ is not completely in the subspace spanned by the relevant documents in the residual collection, however the expanded query vector is closest in angle to the relevant subspace in the residual collection compared to all document vectors in the residual collection.

[7] with the Adi, Cici, Cranfield and Medline data sets [9] (see also [2]) and the Financial Times and Congressional Records from the TREC data sets [1].

4.1.1 Ranking documents

When using relevance feedback only the document vectors in the residual collection A_{n-t} (19), where the t removed document vectors correspond to the t documents used for the feedback cycle, will be scored for relevance.

In order to mimic the optimal scoring (12) we have sorted the documents in relevance order using any of the measures

$$\frac{\|\mathcal{P}_{\tilde{\mathcal{R}}}a_j\|_2}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n-t \tag{20}$$

and

$$\frac{\|a_j - \mathcal{P}_{\tilde{\mathcal{C}}}a_j\|_2}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n-t \tag{21}$$

respectively, where a_j is in the residual collection A_{n-t}.

In the vector model (2) documents are scored measuring angles between the query vector q and each document vector in A. It is natural to score documents measuring angles between the expanded query vectors and each document vector in the residual collection.

$$\frac{(q^+)^T a_j}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n-t, \tag{22}$$

$$\frac{(q^-)^T a_j}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n-t \tag{23}$$

and

$$\frac{(q_{\text{ROCHIO}})^T a_j}{\|a_j\|_2}, \quad j = 1, 2, \ldots, n-t \tag{24}$$

respectively. The vectors (22) – (24) are sorted in descending order.

Note that the three scorings (22) – (24) in general are not equal.

In average, performance is better for all three rankings (22) – (24) using the expanded query vectors compared to the vector model (2).

In figure 1 forming the expanded query vector q^+ we are able to capture all four remaining relevant documents only by considering in total 30 irrelevant documents (compared to the vector model where we had to consider in total 107 irrelevant documents).

The scorings (22) – (24) does not improve average precision for all queries compared to the vector model. Sometimes it is better to use the original

query vector when scoring documents for relevance. In figure 4 the scorings (22) – (24) are compared with the vector model for documents in the residual collection. A few relationships can be noticed. Queries loosing in performance for q^+ also looses in q_{ROCHIO}. Queries where performance improvement is large for q^+ is also large for q_{ROCHIO}.

Retrieval performances for the scorings (20) – (24) and the vector model (2) are compared in figure 5. In average, performance is better for all three expanded query vectors compared to the vector model. Precision is much better but also a small improvement in recall can be seen. The expanded query q^+ gives largest improvement while the q^- score documents rather similar to the original vector model. With other tuning constants for the Rochio queries or the expanded query vector q^- performance might improve. The approximate optimal scoring (20) using the relevant retrieved subspace (14) is also good, while the approximated scoring (21) using the complement (15) is not very effective.

4.2 Explicit restart

Explicit restart means replacing the starting vector q with an improved starting vector and restart the bidiagonalization procedure with this new vector[8]. In eigenvalue computations explicit restart is often used to limit the sizes of the basis set. In the context of IR we want to restart the bidiagonalization procedure with a vector that better captures the connections between the groups of relevant documents. The basic idea is to start the bidiagonalization procedure with the original query vector q and rank the documents using any of the rankings in section 2. Based on relevance feedback information from the user, new improved starting vectors for the bidiagonalization procedure are constructed.

4.2.1 Simple explicit restart

Figure 6 is a recall-precision graph. We constructed expanded query vectors q^+, q^- and q_{ROCHIO} from section 4.1 using relevance feedback. For each query the user judged the 10 top ranked documents from the expanded query measure (6) (using the original query vector q as starting vector in the bidiagonalizing procedure). The bidiagonalization procedures were restarted and

[8]The bidiagonalization procedure is further described in [6]. A summary can be found in appendix A

the documents in the residual collections were ranked using the subspace projection measure (4). Retrieval performances for the different starting vectors are compared in the figure. In general performance is improved by explicit restart.

Even though we measure performance only for the documents in the residual collection, it is important to keep the relevant retrieved documents in the term document matrix when bidiagonalizing. Otherwise performance will decrease[9]. Quite often retrieval performance is further increased by removing the irrelevant retrieved documents from the term document matrix before bidiagonalizing.

4.2.2 Modifying the bidiagonalization procedure

The retrieval performance can be further increased by using the feedback information also when bidiagonalizing. Assume the subspace $\mathcal{E}$ span directions we want to avoid. Starting the bidiagonalizing procedure with any vector orthogonal to $\mathcal{E}$ will start a search orthogonal to the unwanted directions, but it is not enough to guarantee the orthogonality between the basis vectors in Q_{r+1} (3) and $\mathcal{E}$. Technically it is easy to rewrite the BIDIAG procedure to incrementally compute vectors $q_i \in Q_{r+1}$ orthogonal to $\mathcal{E}$, and thus completely avoid all directions in the subspace while bidagonalizing (further details are in appendix B).

Start with $q_1 = \frac{q-\mathcal{P}_{\mathcal{E}}q}{\|q-\mathcal{P}_{\mathcal{E}}q\|_2}$, $\beta_1 = 0$
for $k = 1, 2, \ldots r$ **do**
  $\alpha_k p_k = A^T q_k - \beta_k p_{k-1}$
  $y = Ap_k - \alpha_k q_k$
  $\beta_{k+1} q_{k+1} = y - \mathcal{P}_{\mathcal{E}} y$
end.

The vectors spanning $\mathcal{E}$ need to be chosen with some care. In order not to increment the computational load too much the number of vectors spanning $\mathcal{E}$ needs to be rather moderate. There must be document vectors in A that are not completely in $\mathcal{E}$ otherwise the procedure will vanish.

[9]In order to keep the effect of relevant documents resembling each other in the bidiagonalization procedure it seems to be important that all the relevant document vectors are kept otherwise the resembling effect will be to weak.

$\mathcal{E}$ could for example be chosen to span irrelevant retrieved directions, $\mathcal{E} = \tilde{\mathcal{C}}$ (15). Another option is to let $\mathcal{E}$ be spanned by one vector containing all terms that are in the irrelevant retrieved documents but not in the retrieved relevant. This procedure could also be useful for boolean queries.

4.3 Query subspaces and the union of Krylov subspaces

In this section we will consider query subspaces instead of query vectors. Instead of using the Golub Kahan bidiagonalizing procedure (3) we will use the band Lanczos procedure starting with the query subspace (or subspaces spanned by document vectors corresponding to relevant retrieved documents).

4.3.1 Query subspace

One way to broaden the query is to use query expansion, where more terms are added to the query vector to make it broader. Another way to broaden the query is to let the terms in the query span a subspace $\mathcal{Q}$.

Assume the query vector q consist of s terms, then we let the *query subspace* $\mathcal{Q}$ be spanned by s $m \times 1$ vectors, each vector with one nonzero element corresponding to a term in the query.

Any vector that can be expressed as a linear combination of terms in the query belongs to the query subspace $\mathcal{Q}$, in particular the query vector $q \in \mathcal{Q}$.

As for the query vector (9), the query subspace can be divided into three orthogonal subspaces, $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ where $\mathcal{Q}_1$ is in the relevant subspace (7), $\mathcal{Q}_2$ is in the range of A, $\mathcal{A}$ but not in the relevant subspace and $\mathcal{Q}_3$ is orthogonal to $\mathcal{A}$.

The query subspace can be projected onto the subspace spanned by relevant retrieved document vectors (14) $\mathcal{Q}^+ = \mathcal{P}_{\tilde{\mathcal{R}}}\mathcal{Q}$ or orthogonal to the complement (15) $\mathcal{Q}^- = \mathcal{Q} - \mathcal{P}_{\tilde{\mathcal{C}}}\mathcal{Q}$

We may score the documents for relevancy measuring the angles between the query subspace and each vector in the term document matrix. The cosines of the angle between the query subspace and each document vector in A

$$\|\mathcal{P}_{\mathcal{Q}} a_j\|_2$$

are sorted in decreasing order, and the documents corresponding to the larger cosines are ranked high.

We may also mimic the scorings (22) (23) from section 4.1.1 and score the documents according to the closeness in angles to the projected subspaces respectively

$$\|\mathcal{P}_{\mathcal{Q}^+} a_j\|_2$$

and

$$\|\mathcal{P}_{\mathcal{Q}^-} a_j\|_2.$$

4.3.2 The band Lanczos algorithm

The band Lanczos algorithm [18] (see also [3]) is based on block Krylov subspaces induced by a square matrix B and a block of s linearly independent starting vectors

$$y_1, y_2, \ldots, y_s. \tag{25}$$

The band Lanczos algorithm constructs orthonormal vectors that form a basis for the subspace spanned by the first linearly independent vectors of the block Krylov sequence

$$y_1, y_2, \ldots, y_s, By_1, By_2, \ldots, By_s, B^2y_1, B^2y_2 \ldots$$

If we apply the band Lanczos algorithm to the matrix

$$B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

where A is the $m \times n$ term document matrix, with the starting block

$$Q_s = \begin{bmatrix} q_1 & q_2 & \cdots & q_s \\ 0 & 0 & \ldots & 0 \end{bmatrix} \tag{26}$$

with orthonormal columns that is a basis of the subspace spanned by $y_1, y_2, \ldots, y_s$, it reduces to the BANDL procedure below.

Define $h_{ji} = 0$ when $i < 1$. The following procedure compute the basis matrices Q_{r+s} and P_r and the $(r+s) \times r$ lower $(s+1)$ lower diagonal matrix $H_{r+s,r}$ satisfying

$$H_{r+s,r} = Q_{r+s}^T A P_r. \tag{27}$$

ALGORITHM BANDL(A, Q_s, r)
Start with s orthonormal vectors forming $Q_s = \begin{bmatrix} q_1 & q_2 & \dots & q_s \end{bmatrix}$.
for $j = 1$ **to** r **do**
  $h_{jj}p_j = A^T q_j - p_{j-s}h_{j,j-s} - p_{j-s+1}h_{j,j-s+1} - \cdots - p_{j-s}h_{j,j-s}$
  $w = Ap_j - q_i h_{jj}$
  for $i = j+1$ **to** $j+s-1$ **do**
    $h_{ij} = q_i^T w; w = w - q_i h_{ij}$
  end
  $h_{j+s,j}q_{j+s} = w$
end

With $H_{r+s,r} = \begin{bmatrix} h_{ij} \end{bmatrix}$, the matrix $H_{r+s,r}$ is of size $r+s \times r$ and lower (s+1)-diagonal. In the first part, where computing p_j, previously computed subdiagonal elements or H are used while h_{jj} is computed to give p_j unit norm. In the second part, where computing q_{j+s}, the subdiagonal elements of H are computed as Gram Schmidt orthogonalization coefficients.

Define $Q_{r+s} = \begin{bmatrix} q_1 & q_2 & \dots & q_{r+s} \end{bmatrix}$ and $P_r = \begin{bmatrix} p_1 & p_2 & \dots & p_r \end{bmatrix}$. In exact arithmetic we will have $Q_{r+s}^T Q_{r+s} = I$ and $P_r^T P_r = I$. After $r+s$ iterations the basic relations

$$\begin{aligned} A^T Q_r &= P_r H_{rr}^T \\ AP_r &= Q_{r+s} H_{r+s,r} \end{aligned}$$

will hold. The columns of Q_{r+s} will be an orthonormal basis of the block Krylov subspace $\mathcal{K}_{r+1}(AA^T, Q_s)$ in the document space, spanned by the starting block Q_s and the columns of A.

The columns of P_r similarly span a basis of the block Krylov subspace $\mathcal{K}_r(A^T A, A^T Q_s)$ in the term space spanned by the rows of A. The singular values of $H_{r+s,r}$ will be approximations to those of A.

As for the original bidiagonalizing procedure from section 2 it is easy to rewrite the BANDL procedure to incrementally compute vectors $q_i \in Q_{r+s}$ orthogonal to a subspace $\mathcal{E}$ (section 4.2.2)

4.3.3 The band Lanczos procedure for IR

The BANDL applied to the term document matrix A gives us an opportunity to start with a block of (orthonormal) vectors spanning relevant information.

Let columns in the $m \times s$ matrix Q_s be orthonormal forming a starting block. We apply the BANDL algorithm to the term document matrix A starting at Q_s to receive the two basis matrices Q_{r+s} and P_r and the $r+s \times r$ lower (s+1)-diagonal matrix $H_{r+s,r}$. We let W form an orthonormal basis for the reached subspace spanned by the column vectors in AP_r.

As for the bidiagonalization procedure (3) the reached subspace W, the basis matrices Q_{r+s}, P_r and the $H_{r+s,r}$ matrix are used to score documents in relevance order to the query. A few examples were presented in section 2.

- In the *block subspace projection measure* the documents are sorted according to their closeness measured in angles to the block Krylov subspace $\mathcal{K}_{r+1}(AA^T, Q_s)$. The closer the document is, the more relevant.

$$c_j = \|Q_{r+s}^T a_j\|, \quad j = 1 \dots n. \tag{28}$$

  Sorting the documents according to Q_{r+s} in general give a better scoring than sorting the documents according to Q_{r+1} in the subspace projection measure (4) from section 2.

- For the *block expanded query measure* a projected query vector $\hat{q} = WW^T q$ is constructed using the reached subspace. The documents are sorted using the cosine scoring between $\hat{q}$ and each document vector in A.

$$c_j = \frac{\hat{q}^T a_j}{\|a_j\|}, \quad j = 1 \dots n. \tag{29}$$

  The larger the cosine value the more relevant document.

A few relations should be observed:

If we let the s vectors in the starting block Q_s span be the relevant retrieved subspace $\tilde{\mathcal{R}}$ (14) and stop the iterations in the BANDL procedure when $r = s$, then the block subspace projection measure (28) is the approximate optimal scoring (20) and the block cosine scoring (29) is the cosine of the angle between the expanded query q^+ from section 4.1 and each document vector (22).

If the starting block only consists of the query vector q then the BANDL procedure reduces to the BIDIAG procedure in section 2, otherwise letting $Q_s = \begin{bmatrix} q_1 & q_2 & \dots & q_s \end{bmatrix}$ and using theorem in section 1.3 column vectors in Q_{r+s} and P_r from the BANDL procedure form the union of the s Krylov subspaces $\mathcal{K}_r(AA^T, q_j)$ and $\mathcal{K}_r(A^T A, A^T q_j)$, $j = 1 \dots s$ respectively.

Let the column vectors in Q_s span the relevant retrieved subspace $\tilde{\mathcal{R}}$ (14) The Krylov subspace $\mathcal{K}_r(AA^T, q^+)$ received when using the bidiagonalization procedure from section 2 with A and the expanded starting vector q^+ (16) is a subspace of the block Krylov sequence $\mathcal{K}_{r+s}(AA^T, Q_s)$ received when using the BANDL process with A and the relevant retrieved directions spanned by Q_s, thus we will have $Q_{r+1} \subset Q_{r+s}$, where Q_{r+1} is the basis matrix (3) from the BIDIAG procedure and the Q_{r+s} is the basis matrix (27) from the BANDL procedure.

Used properly the BANDL procedure performs very well. Figure 7 is a recall-precision graph for the block expanded query measure.

4.4 Starting at scoring vector

Sometimes we start the bidiagonalization procedure with a scoring vector p, a weighted combination of documents. Then it is natural to reduce the matrix A to *upper* bidiagonal form by computing orthonormal bases for the Krylov subspaces $\mathcal{K}_r(A^TA, p)$ and $\mathcal{K}_{r+1}(AA^T, Ap)$, using the bidiagonalization procedure

$$\begin{aligned} \rho_k u_k &= Ap_k - \theta_k u_{k-1} \\ \theta_{k+1} p_{k+1} &= A^T u_k - \rho_k p_k \end{aligned}$$

with $k = 1, 2 \ldots r$, $p_1 = p/||p||_2$ and $\theta_1 = 0$. If we start the iteration with $p = A^T q$ this bidiagonalizing procedure can be derived from the BIDIAG procedure discussed in section 2. The relationships between the bidiagonalizations are discussed by Paige and Saunders [16] and also by Golub [11].

5 Numerical experiments

We present our experiments using the Cranfield collection, however the results are general and valid for other sets as well[10]. The overall retrieval performance varies between the sets. For the Medline set retrieval performance is very good while performance for the FT set is more moderate.

[10]Similar experiments were performed using the Adi, Cici, Cranfield and Medline data sets [9] (see also [2]) and the Financial Times and Congressional Records from the TREC data sets [1].

Cranfield is a small collection (1400 documents) with a large number of queries (225 queries). The data set consist of document abstracts in aerodymancis originally used for tests at the Cranfield Institute of Technology in Bedford, England.

We choose to report our experiments with a sequence of figures with appropriate captions.

Preparing the term document matrix We have used a simple term frequency weighting to construct the term document matrix

$$A = [a_{ij}] = \begin{cases} 0 & \text{if term } i \text{ not present in document } j \\ t_{ij} & \text{if term } i \text{ is present in document } j. \end{cases} \tag{30}$$

where t_{ij} is the number of times term i appears in document j. We use one row normalization followed by one column normalization in order to deemphasize common terms and long documents[11]. All rows corresponding to terms appearing in more than 10% of the documents were removed. For a further discussion about weightings for the Krylov subspace method please see [5].

In experiments where relevance feedback is used one initial run is made and the user is shown the top 10 documents. These documents are then used for relevance feedback purposes. Queries where no relevant documents were found among the top 10 or all relevant documents were among the top 10 were removed.

For evaluation measures the *residual collection method* is used. The evaluation of the results compares only to the residual collection A_{n-10}, that is all documents except the ten previously shown to the user are ranked and evaluated. The residual collection method provides an unbiased and realistic evaluation of feedback. However, because highly ranked relevant documents have been removed from the residual collection, there is a risk that the recall-precision figures will be lower than those for standard evaluation methods, and cannot directly be compared.

Relevance is always judged by comparing the results of an algorithm to relevance judgments provided with the test sets. These have been compiled by a panel of human experts who have considered at least all those documents marked as relevant.

[11]The column normalization will destroy the previous row normalization but not completely. Some deemphasizing effect of common terms still remain.

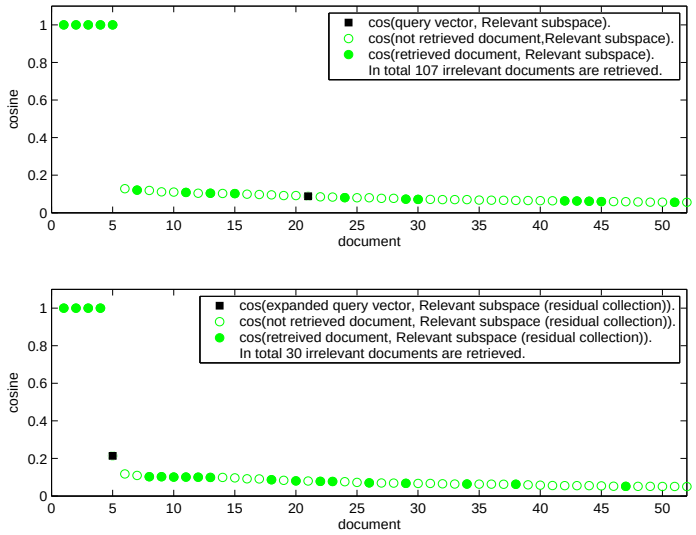

Figure 1: **Upper plot.** Cosines of the angles between the relevant subspace (7) and each document vector and the query vector for query no. 5 from the Cranfield set (only the 50 largest cosines are shown). In order to retrieve all five relevant documents using the vector model(2) 107 irrelevant documents were returned. **Lower plot.** Cosines of the angles between the relevant subspaces in the residual collection (19) and each document vector and the expanded query vector q^+ (16) for query no. 5 from the Cranfield set. (Only the 50 largest are shown). The expanded query vector q^+ was formed using the ten best scored documents from the vector model (2). In order to retrieve all four relevant documents in the residual collection measuring the angles between the expanded query and each document vector in the residual collection 30 irrelevant documents were returned. **In each plot** the cosine for the query vector is marked with a black square. The cosines corresponding to the documents are marked with circles and the documents retrieved in order to capture all the relevant documents are marked with filled circles. Query no. 5 has five relevant documents. Using the vector model one relevant document is scored among the top 10. The query is typical in the sense that in order to retrieve all the relevant documents using the vector model many irrelevant documents are retrieved (making precision rather moderate)

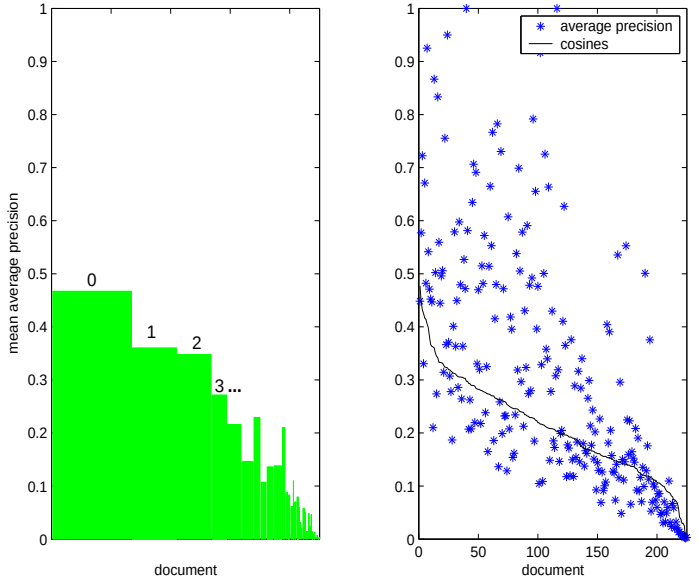

Figure 2: **Left.** Mean average precision for the Cranfield queries when the vector model (2) is used. Each bar is mean average precision when $0, 1, 2 \ldots$ irrelevant documents has smaller angles to its relevant subspace (7) than the query vector. The breadth of each bar is proportional to the number of queries used to compute the mean average precisions. **Right.** Average precision for the 225 Cranfield queries (the stars) when the vector model scoring (2) is used. The queries are sorted according to their closeness to the relevant subspace (7) (the black line). **In each plot** In general average precisions for query vectors orthogonal or with a small part in their relevant subspaces are very moderate. Average precisions for queries closer in angles to their relevant subspaces tend to be better.

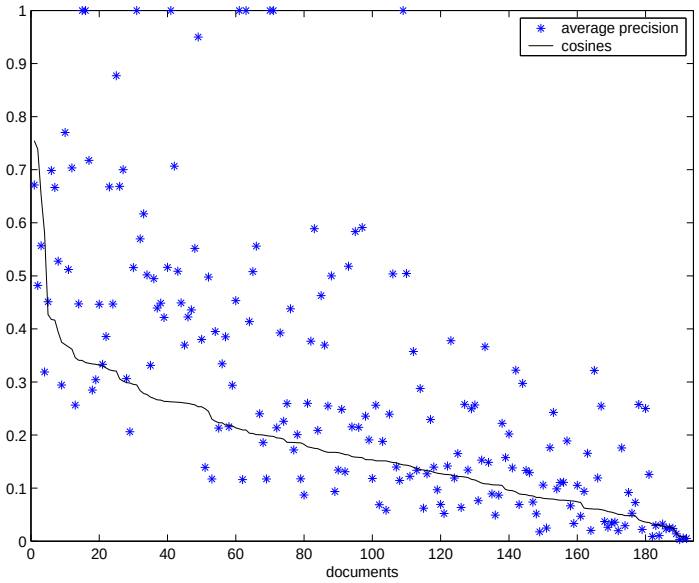

Figure 3: Mean average precisions for the Cranfield queries when the block subspace projection measure (28) is used to score the documents. The queries are sorted according to first principal angle between the subspace $\tilde{\mathcal{R}}$ spanned by relevant retrieved document vectors (14) and the subspace spanned by relevant not retrieved document vectors for each query. Some correlation seems to appear.

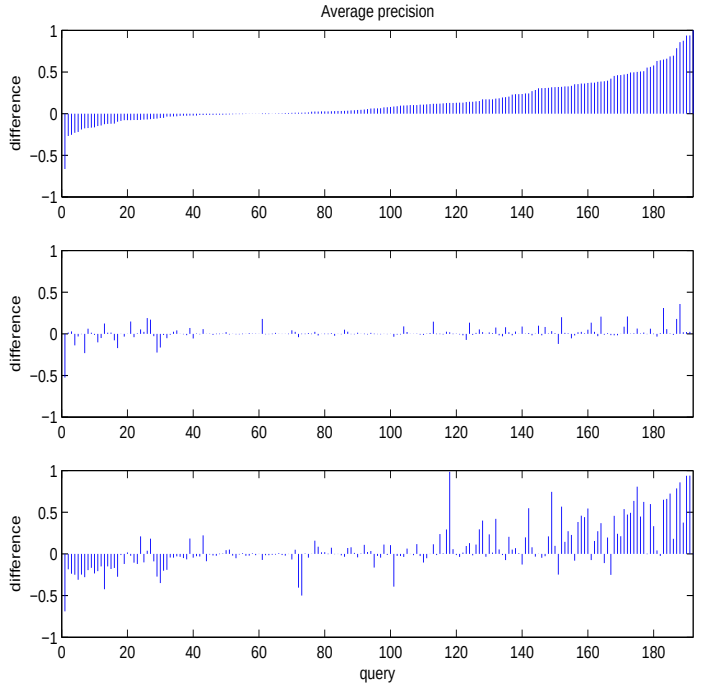

Figure 4: Differences between average precisions for the ranking algorithms (22) (23) (24) using the expanded query vectors from section 4.1 and the vector model (2). Plots from top to bottom are q^+, q^- and q_{ROCHIO}. In all plots the queries are sorted according to the differences in the top plot. Only documents from the residual collection are used when computing average precisions.

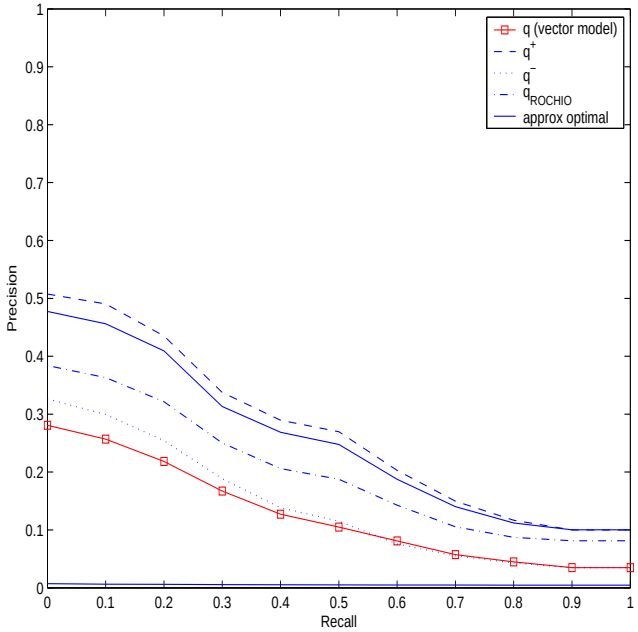

Figure 5: Recall-precision graph for the Cranfield collection comparing the rankings presented in section 4.1.1. The approximate optimal scoring using the relevant retrieved subspace (20) correspond to the upper solid and the approximate optimal scoring using the complement (21) correspond to the lower solid. For all test collections we have tried the q^+ ranking (22) and the approximate optimal scoring (20) are the best. Performance of the q^- (23) and q_{ROCHIO} (24) scorings depend on what tuning constants are used, but performance is rarely above the approximate optimal and the q^+ ranking. All rankings except the approximate optimal scoring (21) give better retrieval performance than the vector model (2).

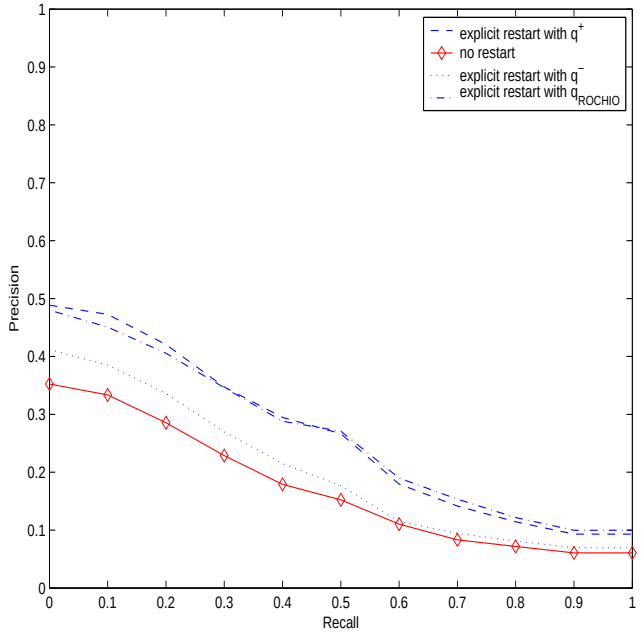

Figure 6: Recall-precision graph for the Cranfield collection. The bidiagonalizing procedure was restarted with the expanded vectors q^+, q^- and q_{ROCHIO} from section 4.1 respectively and the subspace projection measure (4) was used for ranking the documents in the residual collection. For all test collections we have tried explicit restart improves the retrieval performance compared to no explicit restart.

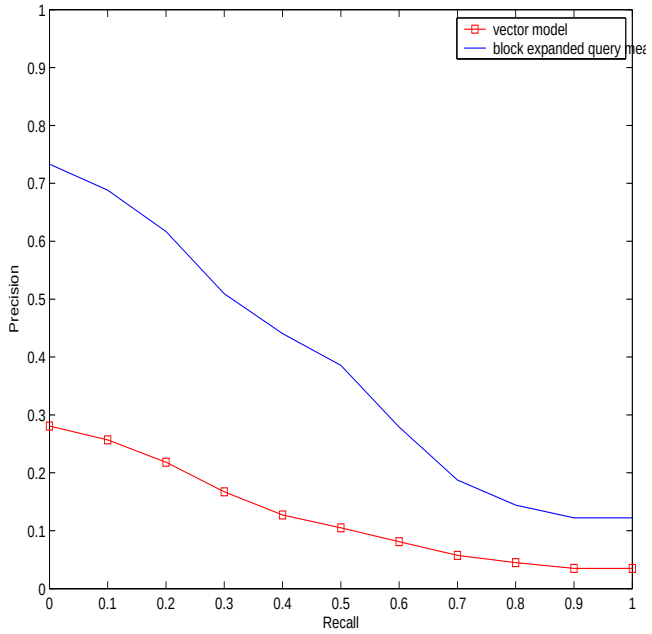

Figure 7: Recall-precision graph for the Cranfield collection. The block expanded query measure (29) is compared with the vector model (2). For all test collections we have tried any of the block measures, block expanded query measure (29) and block subspace projection measure (28), are the best rankings compared to all the other rankings presented in this article.

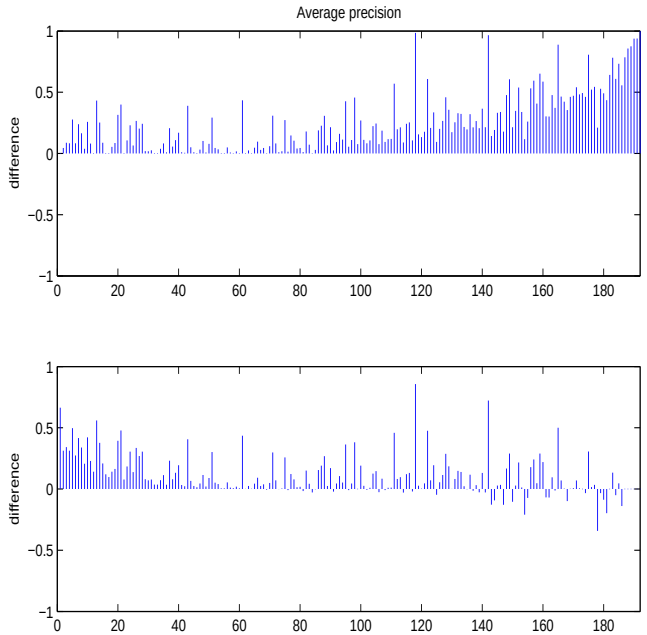

Figure 8: **Upper plot.** Differences in average precisions for block expanded query measure (29) and the vector model (2). **Lower plot** Differences in average precisions for block cosine measure (29) and the q^+ ranking (22) from section 4.1.1. The queries are sorted as in figure (4).

References

[1] *Text REtrieval Conference (TREC)*. `http://trec.nist.gov/`.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, 1999.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[4] Å. Björck and G. H. Golub, *Numerical methods for computing angles between linear subspaces*, Math Comp, 27 (1973), pp. 579–594.

[5] K. Blom, *Experimenting with different weighting schemes for the Krylov Subspace method used for IR*, tech. rep., Dept. of Mathematics, Chalmers university of Technology, 2003.

[6] K. Blom and A. Ruhe, *Information Retrieval using a Krylov Subspace method*, Submitted for publication 2003.

[7] ——, *Information Retrieval using very short Krylov sequences*, in Computational Information Retrieval, M. W. Berry, ed., SIAM, 2000, pp. 39–52.

[8] L. Eldén, *Partial Least Squares vs. Lanczos Bidiagonalization I: Analysis of a Projection Method for Multiple Regression*, Tech. Rep. LiTH-MAT-R-2002-24, University of Linköping, Dept. of Mathematics, 2002.

[9] E. A. Fox, *Characterization of two new experimental collections in computer and information science containing textual and bibliographical concepts*, Tech. Rep. 83-561, `http://www.ncstrl.org`, 1983.

[10] W. B. Frakes and R. Baeza-Yates, *Information Retrieval, Data Structures and Algorithms*, Prentice Hall, 1992.

[11] G. Golub and C. F. van Loan, *Matrix Computations 3 ed.*, Johns Hopkins, 1996.

[12] H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM Journal on Numerical Analysis, 2 (1965), pp. 205–221.

[13] D. Harman, *Relevance feedback and other query modifcation techniques*, in Information Retrieval, Data Structures and Algorithms, W. B. Frakes and R. Baeza-Yates, eds., Prentice Hall, 1992, pp. 241–263.

[14] ——, *A1 (appendix)*, in The Eighth Text REtrieval Conference (TREC-8), D. Harman, ed., NIST Special Publication, 2000, pp. 500–546.

[15] E. Ide, *New experiments in relevance feedback*, in The SMART Retrieval System, G. Salton, ed., Prentice Hall, 1971, pp. 337–354.

[16] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Soft, 8 (1982), pp. 43–71.

[17] J. J. Rochio, *Relevance feedback in information retrieval*, in The SMART Retrieval System – Experiments in Automatic Document Processing, G. Salton, ed., Prentice Hall, 1971.

[18] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Mathematics of Computation, 33(146) (1979), pp. 680–687.

[19] D. S. Watkins, *Fundamentals of Matrix Computations*, John Wiley & Sons, 1991.

[20] J. Xu and W. B. Croft, *Query expansion using local and global document analysis*, Proc. ACM SIGIR, (1996), pp. 4–11.

A The Golub-Kahan bidiagonalization procedure

The Golub-Kahan bidiagonalization procedure is a variant of the Lanczos tridiagonalization algorithm and it is widely used in the numerical linear algebra community.

We start the Golub Kahan algorithm with the normalized query vector $q_1 = q/\|q\|$ and use the term document matrix A, and computes two orthonormal bases P and Q, adding one column for each step k, see [11] in section 9.3.3.

ALOGRITHM BIDIAG(A,q,r):
Start with $q_1 = q/\|q\|$, $\beta_1 = 0$
for $k = 1, 2, \ldots r$ **do**
$\alpha_k p_k = A^T q_k - \beta_k p_{k-1}$
$\beta_{k+1} q_{k+1} = A p_k - \alpha_k q_k$
end.

The scalars α_k and β_k are chosen to normalize the corresponding vectors.

Define

$$\begin{aligned} Q_{r+1} &= \begin{bmatrix} q_1 & q_2 & \ldots & q_{r+1} \end{bmatrix}, \\ P_r &= \begin{bmatrix} p_1 & p_2 & \ldots & p_r \end{bmatrix}, \end{aligned} \tag{31}$$

$$B_{r+1,r} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \alpha_r & \\ & & \beta_{r+1} & \end{bmatrix}.$$

After r steps k we have the basic recursion

$$\begin{aligned} A^T Q_r &= P_r B_r^T \\ A P_r &= Q_{r+1} B_{r+1,r}. \end{aligned}$$

The columns of Q_r will be an orthonormal basis of the Krylov subspace $\mathcal{K}_{r+1}(AA^T, q)$ and the columns of P_r forms an orthonormal basis for the Krylov subspace $\mathcal{K}_r(A^T A, A^T q)$. The lower bidiagonal matrix $B_{r+1,r} = Q_{r+1}^T A P_r$

is the projection of A into these Krylov subspaces and the singular values of $B_{r+1,r}$ will be approximations of those of A.

It is well known [8] that if the query vector q has large components along some singular vectors that do not correspond to the largest singular values of the term document matrix A then the first few basis vectors in Q_{r+1} (31) will contain large components along these singular vectors. If the components in q are not large enough or if the components correspond to the largest singular values then the first basis vectors in Q_{r+1} will be dominated by components from the singular vectors corresponding to the largest singular values.

B The-Golub Kahan bidiagonalization procedure modified

Assume the subspace $\mathcal{E}$ span directions we want to avoid and let the columns of E span the subspace $\mathcal{E}$.

Using the matrix $(I - EE^T)A$ instead of A in the BIDIAG procedure together with a query vector orthogonal to C we get

Start with $q_1 = \frac{(I-EE^T)q}{||(I-EE^T)q||_2}$, $\beta_1 = 0$
for $k = 1, 2, \ldots r$ **do**
$\alpha_k p_k = A^T(I - EE^T)q_k - \beta_k p_{k-1}$
$\beta_{k+1} q_{k+1} = (I - EE^T)Ap_k - \alpha_k q_k$
end.

Noting that $E^T q_k = 0$ for all k, the first row in the loop becomes

$$\alpha_k p_k = A^T q_k - \beta_k p_{k-1}.$$

Since we have $EE^T Ap_k = EE^T(Ap_k - \alpha_k q_k)$ the seco nd row in the loop is equal to the two rows

$$\begin{aligned} y &= Ap_k - \alpha_k q_k \\ \beta_{k+1} q_{k+1} &= y - EE^T y. \end{aligned}$$

Thus it is enough to keep the q_k vectors orthogonal to $\mathcal{E}$. The BIDIAG algorithm can be rewritten to

Start with $q_1 = \frac{(I-EE^T)q}{||(I-EE^T)q||_2}$, $\beta_1 = 0$
for $k = 1, 2, \ldots r$ **do**
$\alpha_k p_k = A^T q_k - \beta_k p_{k-1}$
$y = Ap_k - \alpha_k q_k$
$\beta_{k+1} q_{k+1} = y - EE^T y$
end.