UDC:025.4
Original scientific paper

COMPARISON OF INFORMATION RETRIEVAL TECHNIQUES: LATENT SEMANTIC INDEXING AND CONCEPT INDEXING

Jasminka Dobša[1] Bojana Dalbelo-Bašić[2]
[1]Faculty of Organization and Informatics, Varaždin, CROATIA
jasminka.dobsa@foi.hr
[2]Faculty of Electrical Engineering and Computing, Zagreb, CROATIA
bojana.dalbelo@fer.hr

Abstract: *The task of information retrieval is to extract relevant documents for a certain query from the collection of documents. As large sets of documents are now increasingly common, there is a growing need for fast and efficient information retrieval algorithms. The algorithms we are dealing with are embedded in the vector space model. In this paper we compare two information retrieval techniques: latent semantic indexing and concept indexing*

Keywords: *information retrieval, latent semantic indexing, concept indexing.*

1. INTRODUCTION

The *vector space model* is implemented by creating the *term-document matrix* and a vector of query. Let the list of relevant terms be numerated from 1 to m and documents be numerated from 1 to n. The term-document matrix is an $m \times n$ matrix $A = [a_{ij}]$, where a_{ij} represents the weight of term i in document j. On the other side, we have a query or customer's request. In the vector space model, queries are presented as m-dimensional vectors. The simple vector space model is based on literal matching of terms in the documents and the queries. But we certainly know that literal matching of terms does not necessarily retrieve all relevant documents. Synonyms (more words with the same meaning) and polysemies (words with multiple meaning) are two major obstacles in information retrieval.

The method of LSI was introduced in 1990 [5] and improved in 1995 [4]. It represents documents as approximations and tends to cluster documents on similar topics even if their term profiles are somewhat different. This approximate representation is accomplished by using a low-rank singular value decomposition (SVD) approximation of the term-document matrix. Kolda and O'Leary [12] proposed replacing SVD in LSI by the semi-discrete decomposition that saves memory space. Although the LSI method has empirical success, it suffers from the lack of interpretation for the low-rank approximation and, consequently, the lack of controls for accomplishing specific tasks in information retrieval. The explanation of Latent Semantic Indexing efficiency in terms of multivariate analysis is provided in [2,7,13,15]. A method by Dhillon and Modha [6] uses centroids of clusters

created by the spherical k-means algorithm or so-called *concept decomposition* (CD) for lowering the rank of the term-document matrix. Appling this method, the space on which the term-document matrix is projected is more interpretable. Namely, it is a space spread by centroids of clusters. The information retrieval technique using concept decomposition is called *concept indexing* (CI). Furthermore, the concept decomposition method is computationally more efficient and requires less memory then LSI.

Here we compare SVD/LSI and CD/CI in terms of matrix approximations and precision of information retrieval. A comparison is done on an academic example where vectors of documents and terms are projected on a two-dimensional space (so they can be shown graphically in a plane) and on MEDLINE and CRANFIELD collections. Also, we propose an improvement of CD using the fuzzy *k*-means algorithm and compare this method to the CI method using CD by spherical k-means (CDSKM). We have experimentally shown that the projection of the term-document matrix on centroids obtained by the fuzzy *k*-means algorithm results in a better approximation of the term-document matrix in the sense of the Frobenius norm. Also, we investigate how this improvement in approximation reflects on information retrieval. In [6], it is shown experimentally that centroids achieved by the spherical *k*-means algorithm tend to orthonormality as *k* raises. We will show here that centroids created by fuzzy *k*-means algorithm tend to orthonormality faster.

When we lower the term-document matrix rank, an important question of choice of the right dimension of approximation for the purpose of information retrieval arises. We show here that, when applying CI, there is high correlation between the quality of clustering and mean average precision of information retrieval. This means that the dimension of approximation should be selected according to the natural number of clusters in a specific collection.

The paper is organized as follows. Sections 2 and 3 describe LSI and CI applying CDFKM. In Section 4, we compare these two methods on an academic example. A computational comparison of LSI and CI on a large collection of documents is given in Section 5.

2. THE VECTOR SPACE MODEL AND LSI

Let the $m \times n$ matrix $A = [a_{ij}]$ be the term-document matrix. Then a_{ij} is the weight of the *i*-th term in the *j*-th document. The standard procedure is to normalize the columns of the matrix to be of unit norm. The term-document matrix has an important property of being sparse, i.e. most of its elements are zeros.

A query has the same form as a document; it is a vector, which on the i-th place has the frequency of the i-th term in the query. We never normalize the vector of the query because it has no effect on document ranking. A common measure of similarity between the query and the document is the cosine of the angle between them.

In order to rank documents according to their relevance to the query, we compute $s = q^T A$, where q is the vector of the query and the *j*-th entry in s represents the score in relevance of the *j*-th document.

The LSI method is just a variation of the vector space model. The fundamental mathematical result that supports LSI [10] is that for any $m \times n$ matrix A,, the following singular value decomposition exists:

$$A = U \Sigma V^T \tag{1}$$

where U is the $m \times n$ orthogonal matrix, V is the $n \times n$ orthogonal matrix and V is the $m \times n$ diagonal matrix

$$\Sigma = diag(\sigma_1, \cdots, \sigma_p) \tag{2}$$

where $p = \min\{m,n\}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$. The σ_i are the singular values and u_i and v_i are the *i*-th left singular vector and the *i*-th right singular vector respectively. The second fundamental result [9] is the theorem by Eckart and Young, which states that the distance in the Frobenius norm between A and its k-rank approximation is minimized by the approximation A_k. Here

$$A_k = U_k \Sigma_k V_k^T, \tag{3}$$

where U_k is the $m \times k$ matrix which columns are the first k columns of U, V_k is the $n \times k$ matrix which columns are the first k columns of V, and Σ_k is the $k \times k$ diagonal matrix which diagonal elements are the k largest singular values of A. More precisely,

$$\| A - A_k \|_F = \min_{rank(X) \leq k} \| A - X \|_F . \tag{4}$$

We call A_k *truncated SVD* of A and space spread by columns of U_k k –dimensional *LSI subspace*.

The ranking of documents according to their relevance to the query for the LSI method is executed by calculating the score vector $s = q^T U_k \Sigma_k V_k^T$.

3. CONCEPT DECOMPOSITION

In this section, we describe the concept decomposition by the fuzzy k-means algorithm (CDFKM). The fuzzy k-means algorithm (FKM) [8,16] generalizes the hard k-means algorithm. The goal of the k-means algorithm is to cluster n objects (here documents) in k clusters and find k mean vectors or centroids for clusters. Here we will call these mean vectors *concepts*, because that is what they present. The spherical k-means algorithm used in [6] is just a variation of the hard k-means algorithm, which uses the fact that document vectors (and concept vectors) are of the unit norm.

As opposed to the hard k-means algorithm, which allows a document to belong to only one cluster, FKM allows a document to partially belong to multiple clusters. FKM seeks a minimum of a heuristic global cost function

$$J_{fuzz} = \sum_{i=1}^{k} \sum_{j=1}^{n} \mu_{ij}^b \| x_j - c_i \|, \tag{5}$$

Where $x_j, j = 1, \cdots, n$ are vectors of documents, $c_i, i = 1, \cdots, k$ are concept vectors, μ_{ij} is the fuzzy membership degree of document x_j in the cluster whose concept is c_i and b is a weight exponent of the fuzzy membership. In general, the J_{fuzz} criterion is minimized when concept c_i is close to those documents that have a high fuzzy membership degree for cluster $i = 1, \cdots, k$. By solving a system of equations $\frac{\partial J_{fuzz}}{\partial c_i}$ and $\frac{\partial J_{fuzz}}{\partial \mu_{ij}}$, we obtain a stationary point

$$\mu_{ij} = \frac{1}{\sum_{r=1}^{k} \left(\frac{\| x_j - c_i \|^2}{\| x_j - c_r \|^2} \right)^{\frac{1}{b-1}}}, i = 1,\ldots,k;\ j = 1,\ldots,n \tag{6}$$

$$C_i = \frac{\sum_{j=1}^{n} \mu_{ij}^b x_j}{\sum_{j=1}^{n} \mu_{ij}^b}, \; i = 1,\ldots,k \tag{7}$$

for which the cost function reaches a local minimum.

We will obtain concept vectors starting with arbitrary concept vectors $c_i^0, i = 1,\ldots,k$ and computing fuzzy membership degrees $\mu_{ij}^{(t)}$, cost function J_{fuzz}^t and new concept vectors iterative, where t is the index of iteration, until $\left| J_{fuzz}^{(t+1)} - J_{fuzz}^{(t)} \right| < \varepsilon$ for some threshold ε.

In the special case when, instead of computing $\mu_{ij}^{(t)}$ according to formula (6) in each iteration we put

$$\mu_{ij} = \begin{cases} 1 & \text{if } \|x_j - c_j\| < \|x_j - c_l\| \;\; \forall l \neq i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

we obtain the hard k-means algorithm.

Our target is to approximate each document vector by a linear combination of concept vectors. The *concept matrix* is an $m \times k$ matrix which j-th column is the concept vector c_j, that is

$$C_k = [c_1,\ldots,c_k]. \tag{9}$$

If we assume linear independence of the concept vectors, then it follows that the concept matrix has rank k. Now we define the *concept decomposition* P_k of the term-document matrix A as the least-squares approximation of A on the column space of the concept matrix C_k. Concept decomposition is an $m \times n$ matrix

$$P_k = C_k Z^* \tag{10}$$

where Z^* is the solution of the least-squares problem, that is

$$Z^* = (C_k^T C_k)^{-1} C_k^T A. \tag{11}$$

It can be shown that, for the term-document matrix, rank k approximation obtained by SVD satisfies

$$A_k = U_k \Sigma_k V_k^T = U_k (U_k^T U_k)^{-1} U_k^T A = U_k U_k^T A. \tag{12}$$

So, this approximation is, in fact, the least-squares approximation of matrix A onto the column space of matrix U_k.

4. AN EXAMPLE

In this section we compare the efficiency of LSI and CI by CDFKM on the collection of 15 documents (titles of books), where 9 are from the field of data mining, 5 are from the field of linear algebra and 1 is a combination of these fields (application of linear algebra on data mining). The documents are listed in Table 1. A list of terms is formed from words contained in at least two documents, after which words on the stop list are ejected (conjunctions, articles...) and variations of words are mapped on the same characteristic form (e.g. the terms *matrix* and *matrices* are mapped on the term *matrix*, or *applications* and *applied* are mapped on *application*).

Table 1: Documents and their categorization (DM – data minining documents, LA – linear algebra documents). Document D6 is a combination of the two categories. Words from the list of terms are underlined.

Number	Categorization	Document
D1	DM	Survey of text mining: clustering, classification, and retrieval
D2	DM	Automatic text processing: the transformation analysis and retrieval of information by computer
D3	LA	Elementary linear algebra: A matrix approach
D4	LA	Matrix algebra and its applications in statistics and econometrics
D5	DM	Effective databases for text and document management
D6	Combination	Matrices, vector spaces, and information retrieval
D7	LA	Matrix analysis and applied linear algebra
D8	LA	Topological vector spaces and algebras
D9	DM	Information retrieval: data structures and algorithms
D10	LA	Vector spaces and algebras for chemistry and physics
D11	DM	Classification, clustering and data analysis
D12	DM	Clustering of large data sets
D13	DM	Clustering algorithms
D14	DM	Document warehousing and text mining: techniques for improving business operations, marketing and sales
D15	DM	Data mining and knowledge discovery

As a result, we obtained a list of 16 terms which we have divided in three parts: 8 terms from the field of data mining (*text, mining, clustering, classification, retrieval, information, document, data*), 5 terms from the field of linear algebra (*linear, algebra, matrix, vector, space*) and 3 neutral terms (*analysis, application, algorithm*).

Then we have created a term-document matrix and normalized the columns of it to be of the unit norm. To such a matrix we have applied CDFKM (k=2) and truncated SVD (k=2). Let the truncated SVD be $U_2\Sigma_2V_2^T$ and CDFKM be C_2Z^*. In truncated SVD, rows of U_2 are the approximate (two-dimensional) representation of terms, while rows of V_2 are the approximate (two-dimensional) representation of documents. Here we neglect S_2 part, since S_2 is a diagonal matrix and produces only scaling of the axes. In CDFKM, rows of C_2 are approximate representations of terms and columns of Z^* are approximate representations of documents. Coordinates of terms are listed in Table 2, while coordinates of documents and queries are listed in Table 3. Also, on Figure 1 and 2 images of terms are plotted. From Figure 1 we can see that images of two groups of terms, data mining (DM) terms and linear algebra (LA) terms are grouped together in the case of truncated SVD. In the case of CDFKM, two groups of terms are generally grouped along the axes: along y axis (and near y axis) we have DM terms, and along x axis we have LA terms. Exceptions are terms *information* and *retrieval*. Our assumption is that this is because the model was confused by

D6 document, which contains these terms and LA terms.

We have also created two queries (underlined words are from the list of terms):

1) Q1: Data mining

2) Q2: Using linear algebra for data mining.

For Q1 all data mining documents are relevant, while for Q2 only D6 document is relevant.

Table 2: Coordinates of the terms by SVD and CDFKM

Term	SVD		CDFKM	
	xi	yi	xi	yi
text	0,2093	-0,3075	0,0988	0,4296
mining	0,1613	-0,2876	0,0050	0,4217
clustering	0,2374	-0,4090	0,0796	0,4800
classification	0,1162	-0,1802	0,0000	0,2348
retrieval	0,2652	-0,1997	0,2865	0,1111
analysis	0,2071	-0,0921	0,1874	0,1242
information	0,2077	-0,1090	0,2865	0,0003
linear	0,1855	0,1423	0,2018	0,0012
algebra	0,4960	0,4020	0,5439	0,0023
matrix	0,3700	0,2508	0,4049	0,0012
application	0,1855	0,1423	0,2031	0,0000
document	0,0873	-0,1588	0,0000	0,3185
vector	0,2915	0,1946	-0,3163	0,0011
space	0,2915	-0,1946	0,3163	0,0011
data	0,2495	-0,4110	0,1041	0,4669
algorithm	0,1110	-0,1671	0,1787	0,0699

Table 3: Coordinates of documents and queries by SVD and CDFKM

Document	SVD		CDFKM	
	xi	yi	xi	yi
D1	0,2383	-0,3543	0,0613	0,7377
D2	0,2395	-0,2028	0,3779	0,2564
D3	0,3271	0,2628	0,6919	-0,1369
D4	0,3271	0,2628	0,6928	-0,1378
D5	0,1130	-0,1888	-0,0384	0,5367
D6	0,3435	0,0848	0,7400	-0,0979
D7	0,3479	0,2164	0,7063	-0,0848
D8	0,3356	0,2615	0,7075	-0,1401
D9	0,2245	-0,2538	0,3779	0,2479
D10	0,3356	0,2615	0,7075	-0,1401
D11	0,2182	-0,3127	0,0561	0,6416
D12	0,1855	-0,3320	-0,0054	0,6706
D13	0,1327	-0,2332	0,1086	0,3670
D14	0,1424	-0,2492	-0,0796	0,6914
D15	0,1565	-0,2828	-0,0518	0,6388
Q1	0,2213	-0,3999	-0,0732	0,9033
Q2	0,5668	-0,0291	0,7034	0,7502

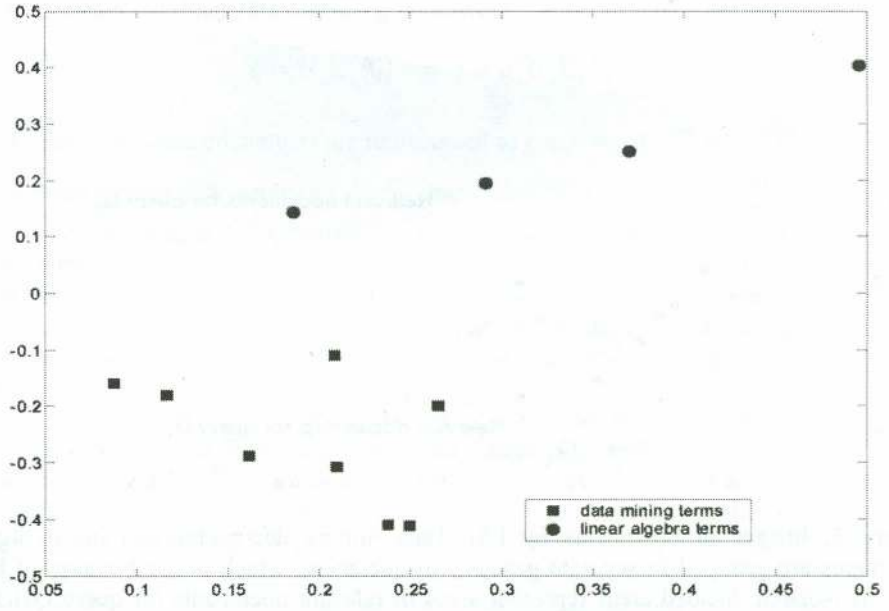

Figure 1: Images of term by LSI. Data mining terms and linear algebra terms are grouped together. y coordinates of data mining terms are negative, while y coordinates of linear algebra terms are positive.

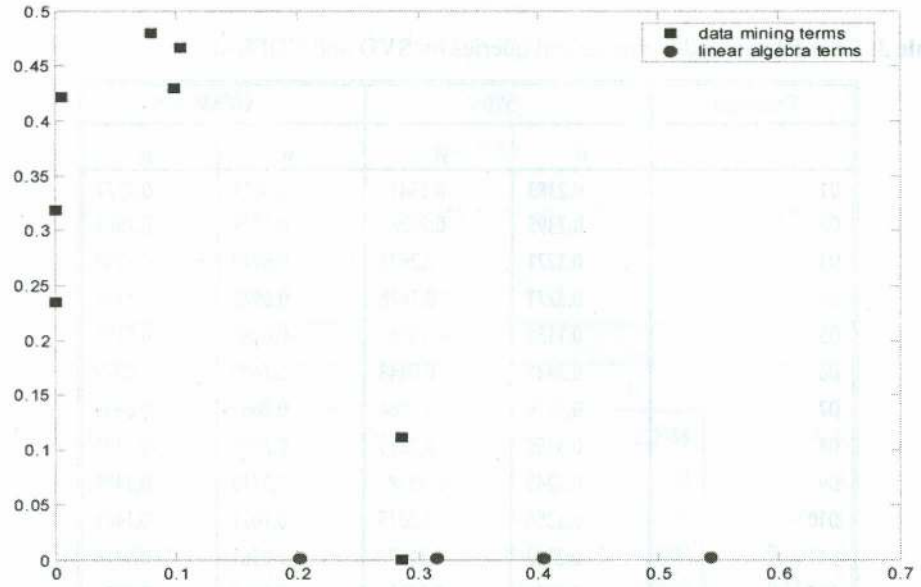

Figure 2: Images of terms by CI. Data mining terms are grouped along y (and near y axis), while linear algebra terms are grouped along x axis. Exceptions are terms *information* and *retrieval.*

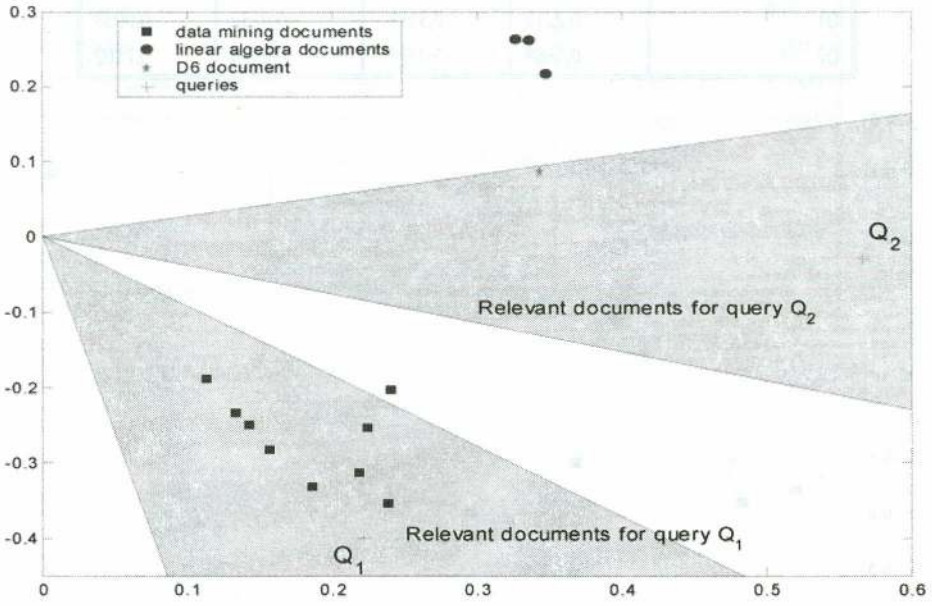

Figure 3: Images of documents by LSI. Data mining documents and linear algebra documents are grouped in separate groups. D6 document, which is combination of these fields is isolated. Shaded areas represent areas of relevant documents for query Q1(Data mining) and query Q2 (Using linear algebra for data mining).

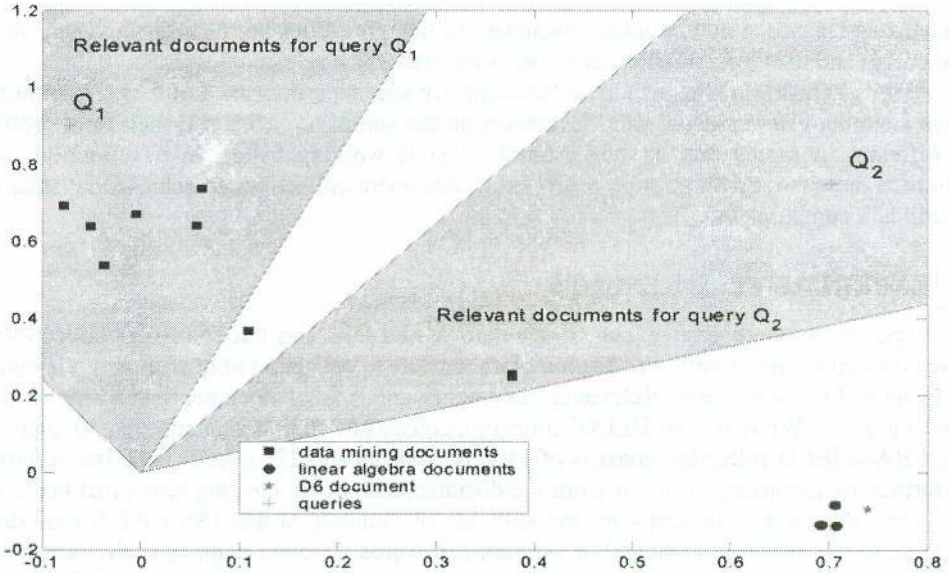

Figure 4: Images of documents by CI. Linear algebra documents form a compact group together with D6 document. Data mining documents are somewhat more dispersed.

Most of the DM documents do not contain words *data* and *mining*. Such documents will not be recognized by the simple term-matching vector space method as relevant. Document D6, relevant for Q2, does not contain any of terms from the list contained in the query. In the vector space model, the query has the same form as the document. Let q be a representation of the query in the vector space model and q its approximate representation using truncated SVD. Then, the following is satisfied:

$$q = U_2\Sigma_2(\tilde{q})^T \Rightarrow \tilde{q} = q^T U_2\Sigma_2^{-1} \tag{13}$$

On the other side, since documents are represented as columns of $Z^* = \left(C_k^T C_k\right)^{-1} C_k^T A$ in CD, the approximate representation of the query by CD will be $\tilde{q} = \left(C_k^T C_k\right)^{-1} C_k^T q$. In Figure 3 and 4, images of approximate representations of documents and queries are plotted. In the SVD projection, DM documents form one group, LA documents another and the D6 document is isolated. In the CD projection, LA documents are grouped; DM documents are somewhat more dispersed, while D6 document is in the group of LA documents. Shaded areas represent the area of relevant documents for queries in the cosine similarity sense.

Now, let us present the results of retrieval. Retrieved documents for query Q1 in descending order, due to their score for the term-matching method, are: D15, D12, D14, D9, D11 and D1. Other documents are not retrieved at all, since their score is 0. So, the term-matching method has retrieved 6 out of 10 relevant documents. The retrieved documents for Q1 applying LSI are: D1, D11, D12, D9, D15, D2, D14, D13, D5 and D6. The score for other documents is much lower and we can state that other documents are not retrieved at all. The retrieved documents are exactly all the relevant documents. The retrieved documents for Q1 applying CI are: D1, D14, D12, D11, D15, D5, D13, D2 and D9. These are all the relevant documents except D6 document. For query Q2, only D6 document is relevant. The term-matching method does not retrieve it at all, the LSI method

recognizes D6 as the most relevant document (although it does not contain any term from the query) and the CI method retrieves D6 as the sixth most relevant document.

As a conclusion of this academic example we can state that the LSI and CI methods have a similar effect; they cluster documents on the similar topic even if their term profile is different. It seems that on this example, LSI is working better. In next section, we compare these two techniques on much larger document collections to achieve statistically significant comparisons.

5. EXPERIMENTAL RESULTS

Experiments were carried out on standard MEDLINE and CRANFIELD collections. Each collection comes with a collection of documents, a collection of queries and relevance judgments for each query. Relevance judgments are lists of documents relevant to the specific query. While the MEDLINE collection consists of 1033 documents and 30 queries, the CRANFIELD collection consists of 1400 documents and 225 queries. The list of terms is formed by extracting all terms from the documents and then ejecting terms that occur in only one document and terms on the stop list of common words (SMART list of stop words). Terms were not stemmed or variations of words were not mapped to the same root form. After this procedure we have obtained a list of 5940 terms for the MEDLINE collection and 4758 terms for the CRANFIELD collection.

TEST A. Firstly, we measure the precision of k-rank approximation P_k obtained by SVD, the concept decomposition by the spherical k-means algorithm (CDSKM) and the concept decomposition by the fuzzy k-means algorithm (CDFKM) for different ranks of approximation k. A common measure is the Frobenius norm of the difference between the term-document matrix and its approximation $\| A - P_k \|_F$. From the theorem of Eckard and Young, we know that the best approximation is obtained by SVD. Here, the emphasis is on the comparison of approximations obtained by CDSKM and CDFKM. From Figures 5 and 6 it is clear that the precision of the k-rank approximation of the term-document matrix is improved by applying CDFKM, compared to applying CDSKM.

TEST B. Secondly, we investigate how the precision of approximation is reflected on the precision of information retrieval. For this comparison, we use the standard measure of *mean average precision* that measures the average precision on standard recall levels [1]. In Tables 4 and 5 and in Figures 7 and 8, a comparison in performance of the LSI method, the CI by CDSKM method and the CI by CDFKM method is shown. We can see that the performance of CI by CDFKM is better than that of LSI and that the performance of CI by CDSKM is the worst, but comparable to the LSI method. In Figure 9 and 10 so called *precision-recall plots* [1] are shown. On precision-recall plots we can see how precision is changing for different levels of recall. It is known [5] that using LSI method precision is improved for higher levels of recall. From Figures 9 and 10 we can see that using CI causes similar effect. Generally, the performance is much better for the MEDLINE collection. For the CRANFIELD collection, the LSI and CI methods do not outperform the simple term-matching method for any rank of approximation; so the application of these methods does not have any sense. For the MEDLINE collection, the best results are obtained by CDFKM method for the rank of approximation of 75 (almost 10% better MAP then by the term-matching method). In this case, documents are presented in a 75×1033 matrix instead of a 5940×1033 matrix, as in the case of the simple vector space model. Anyway, this is not such a significant saving of memory space as it seems at first sight, since the term-

document matrix is sparse, but the representations of documents by LSI or CI generally are not. In Table 4 and 5, we list memory spaces required for matrices that represent documents. The starting term-document matrix is stored in a sparse form, while compressed representations are stored as double precision floats.

TEST C. Here we measure the average inner product between concept vectors cj, $j = 1,\ldots,k$ as

$$\frac{2}{k(k-1)} \sum_{j=1}^{k} \sum_{l=j+1}^{k} c_j^T c_i^T \tag{14}$$

The average inner product takes values in interval [0,1], where smaller values correspond to concept vectors whose average angle between them is close to p/2.

Table 4: Mean Average Precision and memory space required to store documents of the MEDLINE collection for LSI and CI (CDSKM and CDFKM) methods. The best results for every method are bolded.

k	CDSKM	CDFKM	LSI	Memory space (KB)
25 50 75 100 125 150 175 200 225 250	36,61 44,28 44,09 44,55 **45,14** 42,87 42,68 44,05 44,70 44,09	41,71 50,60 **53,13** 48,96 49,58 49,81 49,83 49,50 49,33 49,33	40,23 47,79 **48,59** 48,58 47,68 47,35 47,08 46,62 46,34 45,82	202 404 605 807 1009 1221 1412 1614 1816 2018
Term-matching	43,54	43,54	43,54	616

Table 5: Mean Average Precision and memory space required to store documents of the CRANFIELD n for the LSI and CI (CDSKM and CDFKM) methods. The best results for every method are bolded.

k	CDSKM	CDFKM	LSI	Memory space (KB)
25 50 75 100 125 150 175 200 225 250	9,66 12,52 14,08 14,85 16,33 16,21 16,98 17,35 **17,95** 17,42	9,58 14,00 15,60 17,44 17,76 17,92 18,34 19,77 **19,87** 19,25	9,59 12,71 14,74 16,02 16,93 17,74 18,22 **18,73** **18,73 18,73**	273 547 820 1094 1367 1641 1914 2188 2461 2734
Term-matching	20,89	20,89	20,89	924

Table 6: Correlation matrix for the MEDLINE collection

	k	MAP	J_{fuzz}
k	1,0000	0,7025	-0,8311
MAP	0,7025	1,0000	-0,9682
J_{fuzz}	-0,8311	-0,9682	1,0000

Table 7: Correlation matrix for the CRANFIELD collection

	k	MAP	J_{fuzz}
k	1,0000	0,9145	-0,9044
MAP	0,9145	1,0000	-0,9883
J_{fuzz}	-0,9044	-0,9883	1,0000

From Figures 11 and 12 we can see that concept vectors obtained by the fuzzy k-means algorithm tend to orthonormality faster then those obtained by the spherical k-means algorithm, particularly for the MEDLINE test collection.

TEST D. From Figures 7 and 8, we see that the mean average precision (MAP) obtained by the LSI method depends on the rank of approximation being more stable than the MAP achieved by the CI method. Now we examine if there is correlation between MAP and the quality of clustering for the CI method. In other words, would MAP be better if we chose the rank of approximation to be a natural number of clusters for a specific collection? We will take the cost function J_{fuzz} given in equation (5) as a measure of the quality of clustering. J_{fuzz} is generalized within the groups sum of square errors function and will take smaller values if the number of clusters k is chosen to be the natural number of clusters in the collection. It is obvious that growth of rank of approximation generally causes growth of MAP and drop of J_{fuzz}. We have also calculated correlations between MAP and the rank of approximation and J_{fuzz} and the rank of approximation to test if MAP and J_{fuzz} are directly correlated. Correlations are calculated based on 46 observations for the rank of approximation $k \in [100,1]$ and they are listed in Table 6 and Table 7. All correlations are on the level of significance $p << 0,01$. From the correlation matrices we see that correlations between MAP and J_{fuzz} take the highest absolute values for both collections, meaning that those two characteristics are directly correlated. That is a statistical confirmation of the intuition that the number of clusters should be chosen according to the large enough number of clusters in the collection

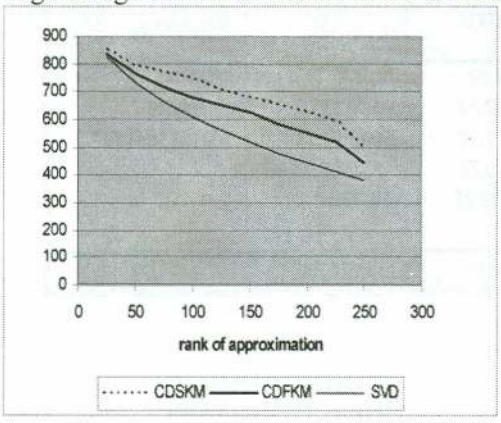

Figure 5: Comparison of approx. errors $\| A - P_k \|_F$ – for MEDLINE collection

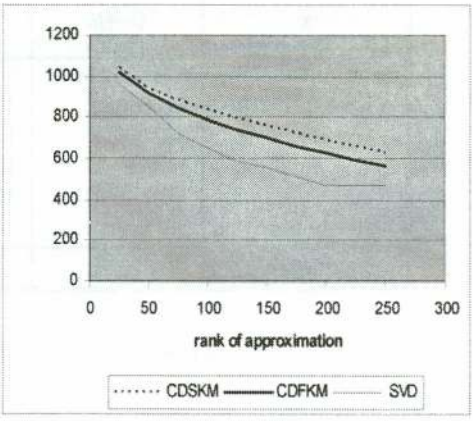

Figure 6: Comparison of approx. errors $\| A - P_k \|_F$ – for CRANFIELD collection

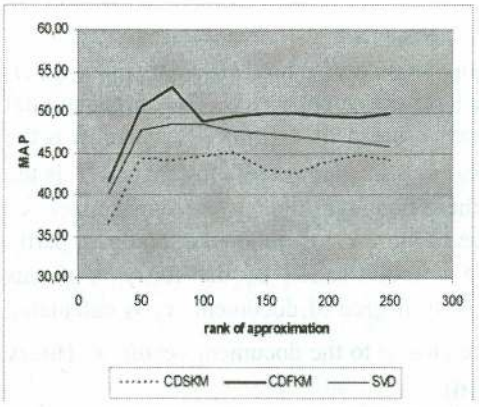

Figure 7: Mean average precision for MEDLINE collection

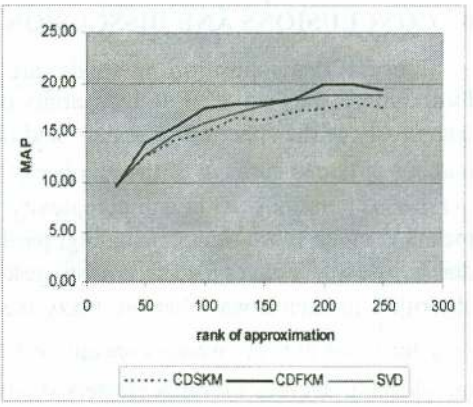

Figure 8: Mean average precision for CRANFIELD collection

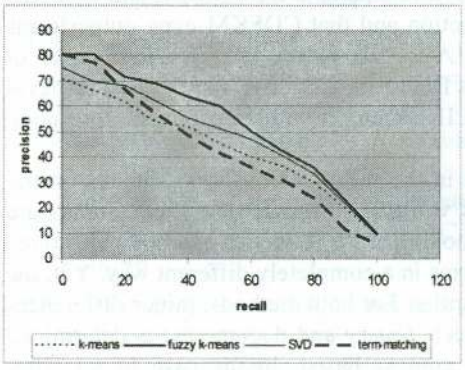

Figure 9: Precision-recall plot for MEDLINE collection

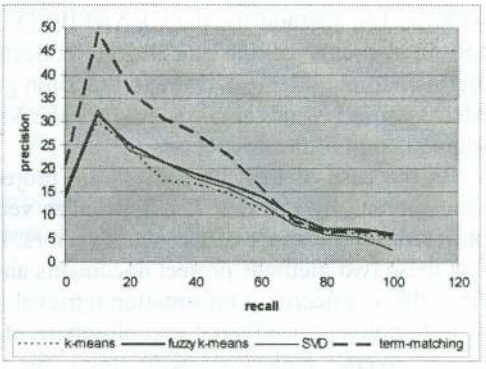

Figure 10: Precision-recall plot for CRANFIELD collection

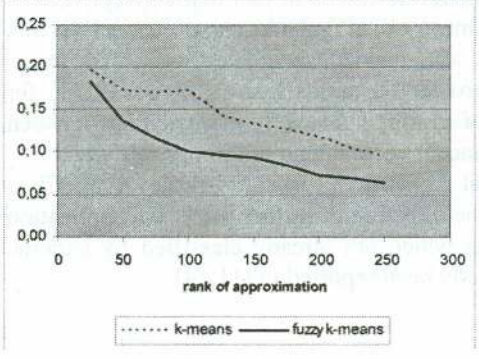

Figure 11: Average scalar product between between concept vectors for MEDLINE collection

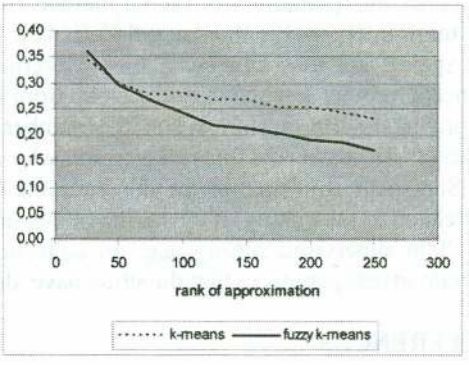

Figure 12: Average scalar product between concept vectors for CRANFIELD collection

6. CONCLUSIONS AND DISSCUSION

Concept decomposition methods are computationally more efficient then SVD. Furthermore, they can exploit the sparcity of the term-document matrix. The computational complexity of the spherical k-means and of fuzzy k-means is $O(nmkT)$, where n is the number of documents, m is the number of terms, k is the number of clusters and T is the number of iterations. Although complexity for these two algorithms is the same, fuzzy k-means is much more time consuming, partly due to more computational operations, partly due to slower convergence. We suggest here a modification of the fuzzy k-means algorithm in such a way that the fuzzy membership degree of document x_j is calculated only for those clusters whose concept vectors are closest to the document vector x_j (fuzzy membership degrees for other clusters should be 0).

The fact that matrix approximations obtained by CDSKM and CDFKM are less precise than by SVD does not reflect on the precision of information retrieval for the two standard collections we have applied. We see that the MAP is comparable for CDSKM and CDFKM with the LSI method for the CRANFIELD collection and that CDFKM even outperforms LSI in the case of the MEDLINE collection. Also, we notice that, for low ranks of approximation, the mean average precision grows fastest for CDFKM. In [5], good retrieval results using LSI are also reported for the MEDLINE data set with an explanation of good segmentation of the collection.

In the case of LSI, documents are projected in the means of the least squares on the space spread by the first k left singular vectors while, in the case of CI, documents are projected on the space of k concept vectors. By looking at the academic example, we notice that these two methods project documents and terms in a completely different way. Yet, the final effect concerning information retrieval is similar. For both methods, minor differences in terminology are ignored and closeness of objects (query and documents) is determined by the overall pattern of term usage, so it is context based. In the case of CI, after projection, documents are presented as a linear approximation of concept vectors, terms are substituted intuitively by concepts, which are representatives of sets of terms. The reason for better interpretability of the CI method compared to LSI is in fact that concept vectors are more interpretable then singular vectors. Contrary to singular vectors, concept vectors are sparse and they can be labeled by terms, which have the greatest weight in them. Concept vectors have entries different from zero for the terms that are characteristic for belonging to the cluster. When the number of clusters grows, term matching between concepts decreases and this is the reason why concept vectors tend to orthogonality.

Statistical confirmation of the intuition that MAP of information retrieval by CI is correlated to the quality of clustering points in the direction of further work: the application of CI in supervised setting, e.g. on collections which are already classified by experts. Certain investigations in that direction have already been reported in [11,14].

REFERENCES

[1] R. Baeza-Yates, B.Ribeiro-Neto. *Modern Information Retrieval*, Addison-Wesley, ACM Press, New York, 1999.

[2] B.T. Bartell, G.W. Cottrell, R.K. Belew. Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling, *SIGIR-1992*, pp. 161-167.

[3] M. W. Berry, Z. Drmač, E. R. Jessup. Matrices, Vector Spaces and Information Retrieval, *SIAM Review*, Vol. 41, No. 2, 1999, pp. 335-362.

[4] M. W. Berry, S. T. Dumais, G. W. O'Brien. Using linear algebra for intelligent information retrieval, *SIAM Rewiew,* Vol. 37, 1995, pp. 573-595.

[5] S. Deerwaster, S. Dumas, G. Furnas, T. Landauer, R. Harsman. Indexing by Latent semantic analysis, *J. American Society for Information Science,* Vol. 41, 1990, pp. 391-407.

[6] I. S. Dhillon, D. S. Modha. Concept Decomposition for Large Sparse Text Data using Clustering, *Machine Learning* , Vol. 42, No. 1, 2001, pp. 143-175.

[7] C.H.Q. Ding. A Similarity-based Probability Model for Latent Semantic Indexing, *SIGIR1999*, pp. 58-65.

[8] R.O. Duda, P.E. Hart, D.G. Stork. *Pattern Classification*, second edition, Willey, New York, 2001.

[9] C. Eckart, G. Young. The approximation of one matrix by another of lower rank, *Psychometrica,* Vol. 1, 1936, pp. 211-218.

[10] G. H. Golub, C. F. Van Loan. *Matrix Computation*, The John Hopkins University Press, Baltimore, Maryland, 1996.

[11] G.Karypis, E. Hong. Conccept Indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization, Technical Report TR-00-016, Department of Computer Science, University of Minnesota, Minneapolis, 2000. Available on the WWW at URL *http://www.cs.umn.edu/~karypis*.

[12] T. Kolda, D. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing in information retrieval, *ACM Trans. Inform. Systems,* Vol. 16, 1998, pp. 322-346.

[13] C. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis, *Journal of Computer and System Sciences*, Vol. 61, No. 2, 2000, pp. 217-235.

[14] H. Park, M. Jeon, J. Ben Rosen. Lower Dimensional Representation of Text Data based on Centroids and Least Squares, *BIT,* Vol. 43, No. 2, 2003, pp. 1-22.

[15] R.E. Story. An Explanation of the Effectiveness of Latent Semantic Indexing by Means of a Bayesian Regression Model, *Information Processing & Management*, Vol. 32, No. 3, pp. 329-344.

[16] J. Yen, R. Langari. *Fuzzy Logic: Intelligence, Control and Information,* Prantice Hall, New Jersey, 1999.

Received: 10 March 2003
Accepted: 25 September 2004