Optimum Polynomial Retrieval Functions (Extended Abstract)

Norbert Fuhr
Technische Hochschule Darmstadt, Fachbereich Informatik
Karolinenplatz 5, D-6100 Darmstadt / West Germany

Abstract

We show that any approach to develop optimum retrieval functions is based on two kinds of assumptions: first, a certain form of representation for documents and requests, and second, additional simplifying assumptions that predefine the type of the retrieval function. Then we describe an approach for the development of optimum polynomial retrieval functions: request-document pairs (f_l, d_m) are mapped onto description vectors $\vec{x}(f_l, d_m)$, and a polynomial function of the form $\vec{a}^T \cdot \vec{v}(\vec{x})$ is developed such that it yields estimates of the probability of relevance $P(R|\vec{x}(f_l, d_m))$ with minimum square errors. We give experimental results for the application of this approach to documents with weighted indexing as well as to documents with complex representations. In contrast to other probabilistic models, our approach yields estimates of the actual probabilities, it can handle very complex representations of documents and requests, and it can be easily applied to multi-valued relevance scales. On the other hand, this approach is not suited to log-linear probabilistic models, and it needs large samples of relevance feedback data for its application.

1 Introduction

A major goal of IR research is the development of effective retrieval methods. Any efforts in this direction are based (explicitly or implicitly) on a certain concept of optimum retrieval. In contrast to perfect retrieval (e.g. retrieve all relevant documents ahead of the first nonrelevant one), optimum retrieval is defined with respect to certain realistic restrictions concerning the retrieval process. Furthermore, some evaluation criteria must be given. Then optimum retrieval is the best retrieval (in terms of the evaluation criteria) that can be achieved while following the predefined restrictions.

With the exception of a recent approach ([Wong et al. 88], see section 4) only for probabilistic IR models optimum retrieval has been defined precisely and the optimality has been proved theoretically: the "Probability Ranking Principle" (PRP) described in [Robertson 77] says that optimum retrieval is achieved when documents are ranked according to decreasing values of their probability of relevance (with respect to the current request), $P(R|f_k, d_m)$, where R denotes the event that a request-document pair (f_l, d_m) is judged relevant by the user.

The explanation of the PRP given in [Robertson 77] and later discussions of this topic lack a precise definition of the events the probabilities relate to. In our view, these probabilities relate to the system's representations of documents and requests, not to the documents and requests itself. For example, many retrieval models take sets of terms as representations of documents and queries. We will denote a specific request with $\underline{f_l}$ and a specific document with $\underline{d_m}$ while f_l and d_m stand for the corresponding representations. This distinction also makes the difference between perfect and optimum retrieval clearer: Because of the representation, the system's knowledge about documents and requests is limited, and it cannot distinguish between different objects which have the same representation. Therefore, perfect retrieval is not possible for a real system. This statement is not in conflict with the observation that there are hardly ever two objects in existing collections which share the same representation: We regard collections as samples from possibly infinite sets of documents and requests, where there may be several (up to infinity) objects with the same representation. Because of the poor representation of retrieval objects that is actually in use (in comparison to the human understanding of documents

and requests), only a probabilistic approach is adequate for these representations. Approaches that attempt to perform perfect retrieval for the current collection will fail as soon as the next document or request is added to the collection.

Regarding probabilistic IR models in more detail, it becomes obvious that these models do not really estimate probabilities for each specific representation: they use additional simplifying assumptions in order to approximate these probabilities. As an example, regard models which use sets of features (e.g. terms) as representations. In practice, it is impossible to estimate the correct probability for a specific set of features, because in most cases there will not be any relevance information available for this set. Therefore, probabilistic models need additional independence assumptions: this way, probability estimates can be performed for single features (or pairs/triplets), and the probability for a set is computed by combining the probabilities of the elements.

This two kinds of approximations can be found with any probabilistic IR model: First is the choice of a certain form of representation for documents and requests. With respect to this representation, optimum retrieval (according to the PRP) can be defined. Second is the selection of an approximation method (e.g. certain independence assumptions).

For the least squares polynomial (LSP) approach described in this paper, both kinds of approximations are only roughly defined and can be optimized with respect to a specific application. This approach is based on the representation of request-document relationships in vectorial form $\vec{x}(f_l, d_m)$. Then a polynomial function $a(\vec{x})$ is developed which yields estimates of $P(R|\vec{x})$. Both the mapping of a request-document relationship onto the so-called "description vector" and the class of the polynomial function (e.g. linear, quadratic) can be chosen.The LSP approach is described in detail in the following section.

2 Least squares polynomial retrieval functions

The LSP approach originally has been developed in the context of pattern recognition methods [Schürmann 77] as a refinement of general minimum square error procedures [Duda & Hart 73, pp. 151–159]. In [Knorz 83b] experiments with LSP indexing functions are described and the application for the development of retrieval functions is proposed.

The task of the LSP approach is to classifiy objects into one of $n+1$ classes. In the retrieval case, the objects are request-document pairs $(\underline{f}_l, \underline{d}_m)$, and the classification is the assignment of a value r_k from a relevance scale $R_n = (r_0, \ldots, r_n)$ (The case of multi-valued relevance scales is discussed in more detail in the following section). In order to develop a LSP retrieval function, a mapping of request-document relationships onto description vectors $\vec{x} = \vec{x}(f_l, d_m)$ has to be defined. Furthermore a representative sample of request-document pairs together with their relevance judgements must be given. From this sample, a function is derived which yields for a pair $(\underline{f}_l, \underline{d}_m)$ estimates of the probabilities $P(r_k|\vec{x}(f_l, d_m)), k = 0 \ldots n$.

In the following, the relevance judgement r_k of a pair (f_l, d_m) is represented by a vector $\vec{y} = (y_0, \ldots, y_n)$ with

$$y_i = \begin{cases} 1, & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases}$$

Now we seek for a vectorial regression function $\vec{e}_{opt}(\vec{x})$ which yields optimum approximations $\hat{\vec{y}}$ of the class vectors $\vec{y}$. As optimizing criterion we use the condition ($E(.)$ denotes the expectation):

$$E(|\vec{y} - \vec{e}_{opt}(\vec{x})|^2) \stackrel{!}{=} \min. \qquad (1)$$

With this condition, $\vec{e}_{opt}(\vec{x})$ yields probabilistic estimates $P(r_k|\vec{x})$ in the components of $\hat{\vec{y}}$ (see [Schürmann 77, pp. 163–164]).

Equation (1) is the formulation of a general variation problem: Among all possible functions $\vec{e}(\vec{x})$, we seek for the optimum function fulfilling the above criterion. Because of the probabilistic nature of the approach, $\vec{e}_{opt}(\vec{x})$ is the optimum retrieval function (with respect to the chosen vectorial representation of request-document relationships) that follows the PRP.

However, this variation problem cannot be solved in its general form, so we have to restrict the search to a predefined class of functions. With this restriction, we will only find a sub-optimum solution, but our general variation problem now becomes a parameter optimization task. The resulting functions yield least squares approximations of $\vec{e}_{opt}$: In [Schürmann 77, pp. 178–180], it is shown that an approximation with respect to the expression $E(|\vec{y} - \hat{\vec{y}}|^2) \stackrel{!}{=} \min$ yields the same result as an optimization fulfilling the condition $E(|E(\vec{y}|\vec{x}) - \hat{\vec{y}}|^2) \stackrel{!}{=} \min$. So our restricted optimiziation process yields least squares approximations of the probabilities $P(r_k|\vec{x}(f_l, d_m))$.

In the LSP approach, polynomials with a predefined structure (see below) are taken as function classes. Let $\vec{v} = \vec{v}(\vec{x})$ be a class of polynomials. The task of the parameter optimization now is to estimate $n+1$ coefficient vectors $\vec{a}_k (k = 0 \ldots n)$ such that the average (as approximation of the expectation) $\overline{|\vec{y} - \vec{a}_k{}^T \cdot \vec{v}|^2}$ is minimized. By assembling the coefficient vectors $\vec{a}_k$ in a coefficient matrix

$A = (\vec{a}_0, \vec{a}_1, \ldots, \vec{a}_n)$, the optimizing criterion can be formulated as $E(|\vec{y} - A^T \cdot \vec{v}|^2) \stackrel{!}{=} \min$. The class of polynomials $\vec{v}(\vec{x})$ is specified as follows: for each relevance class r_k we define a polynomial

$$\begin{aligned} v_k(\vec{x}) &= a_{0k} + a_{1k} \cdot x_1 + + a_{2k} \cdot x_2 + \ldots \\ &\quad + a_{N+1,k} \cdot x_1^2 + a_{N+2,k} \cdot x_1 \cdot x_2 + \ldots \end{aligned}$$

where N is the number of dimensions of $\vec{x}$. So the class of polynomials is given by the components $x_i^l \cdot x_j^m \cdot x_k^n \cdot \ldots$ $(i, j, k \in [1, N]; l, m, n \geq 0)$ which are to be included in the polynomial. In practice, mostly linear and quadratic polynomials are regarded.

The coefficient matrix is computed by solving the equation system [Schürmann 77, pp. 175–176]

$$E(\vec{v} \cdot \vec{v}^T) \cdot A = E(\vec{v} \cdot \vec{y}). \qquad (2)$$

The development of a polynomial retrieval function is performed in two steps:

1. Statistical evaluation of a learning sample: A representative sample of request-document relationships together with relevance judgements must be given. From these relationships, the pairs $(\vec{x}, \vec{y})$ are derived and then the empirical momental matrix M is computed:

$$M = \begin{pmatrix} \overline{\vec{v} \cdot \vec{v}^T} & \overline{\vec{v} \cdot \vec{y}^T} \\ \overline{\vec{y} \cdot \vec{v}^T} & \overline{\vec{y} \cdot \vec{y}^T} \end{pmatrix}.$$

   The matrix M contains both sides of the equation system (2).

2. Computation of the coefficient matrix by means of the Gauss-Jordan algorithm [Schürmann 77, pp. 199–227].

There are two important properties of the LSP retrieval functions:

- The basic assumptions of the LSP approach is that we can approximate the expectations by average values. In order to avoid parameter estimation problems, learning samples of sufficient size are required. From previous experience with LSP applications in the IR context [Knorz 83a], [Fuhr 88], we can derive a rule of thumb for this size: per component of the coefficient vector $\vec{a}_k$, there should be about 50–100 elements in the learning sample. Therefore, it seems to be inappropriate to develop request-specific retrieval functions. Instead, we define request-independent retrieval functions by mapping request-document pairs $(\underline{f}_j, \underline{d}_m)$ onto description vectors $\vec{x}(f_l, d_m)$.
- LSP retrieval functions yield estimates of the probability of relevance $P(r_k|\vec{x}(f_l, d_m))$, in contrast to other probabilistic models where it is nearly impossible to get these estimates, because there are too many parameters which can be hardly estimated. We think that estimates of the probability of relevance could help a user of an IR system to get some impression of the overall quality of an answer.

3 LSP retrieval functions and probabilistic models

In this section, we first describe the development of LSP retrieval functions for non-binary relevance scales, then we discuss the relationship between LSP and log-linear models.

Bookstein [Bookstein 83] has shown how the PRP can be extended to non-binary, ordinal relevance scales: Assume that for the $n+1$ relevance values with $r_o < r_1 < \ldots < r_n$ the corresponding costs for the retrieval of a document with that relevance judgement are $C_0 > C_1 > \ldots > C_n$. Then documents should be ranked according to their expected costs

$$EC(f_l, d_m) = \sum_{k=0}^{n} C_k \cdot P(r_k|f_l, d_m)$$

Now we apply the LSP approach for the estimation of the probabilities $P(r_k|\vec{x}(f_l, d_m))$:

$$P(r_k|\vec{x}) \approx \sum_{j=0}^{L} \alpha_{jk} \cdot v_j(\vec{x})$$

(where L is the number of dimensions of $\vec{v}(\vec{x})$). Then the expected costs can be approximated by

$$\begin{aligned} EC(\vec{x}) &\approx \sum_{k=0}^{n} C_k \left(\sum_{j=0}^{L} \alpha_{jk} \cdot v_j \right) \\ &= \sum_{j=0}^{L} \left(\sum_{k=0}^{n} C_k \cdot \alpha_{jk} \right) v_j \end{aligned}$$

From the above expression, we can see that we only need a single polynomial instead of $n+1$ polynomials for the different relevance classes. This single polynomial yields estimates of the expected costs, which is our ranking criterion. For the coefficients of this polynomial, the following relationship holds:

$$a_j = \sum_{k=0}^{n} C_k \cdot \alpha_{jk}$$

In order to develop this kind of polynomial, we have to change the definition of the class vector $\vec{y}$: now this vector has only one component, to which we assign the appropriate cost value C_k for

a request-document relationship with relevance judgement r_k. This way, the polynomial yields with $\hat{y}(\vec{x})$ a least squares approximation of $E(y|\vec{x}) = EC(\vec{x})$. Most probabilistic ranking functions can be expressed in a log-linear form (e.g. [Rijsbergen 79], [Robertson et al. 82]) similar to

$$\log P(R|\vec{x}) = a_0 + \sum_{i=1}^{m} a_i \log p_i$$

where the a_i's are some coefficients (mostly one of $\{-1, +1\}$) and the p_i's are single probabilities, products or quotients of probabilities. This log-linear form seems to be very well suited for the application of the LSP approach, in order to estimate coefficients (a_i) or probabilistic parameters ($\log p_i$).

For a more detailed discussion of this approach, let us make the simplifying assumption that the components of the description vector are assigned the values $x_i = \log p_i$. Furthermore we will only regard a single polynomial for the estimation of $P(R|\vec{x})$. Instead of the class value $y \in \{0, 1\}$, we would now have to take the value $\log y$, where we must define a suitable value for $\log 0$. With the LSP retrieval function, we would try to approximate

$$\log P(R|\vec{x}) = \log E(y|\vec{x}) \approx \vec{a}^T \cdot \vec{x}$$

However, this approach cannot work, because the optimizing criterion for the LSP now is

$$E(|\log y - \vec{a}^T \cdot \vec{x}|^2) \stackrel{!}{=} \min$$

from which we get a least squares approximation of $E(\log(y|\vec{x}))$, but

$$E(\log(y|\vec{x})) \neq \log E(y|\vec{x})$$

So this kind of LSP retrieval function does not yield the desired estimates. This is a paradox situation: the favourable property of the LSP retrieval function to yield estimates of the probability of relevance is here in conflict with the probabilistic model. The probabilistic model suggests a log-linear form, while the LSP approach only can handle polynomials.

An alternative method of developing log-linear functions by using maximum-likelihood estimates has been published recently [Bookstein 88].

4 Optimum linear retrieval

In [Wong et al. 88], Wong et al. give an optimum linear retrieval function for the vector model [Salton 71]. There is earlier work in this area done by Rocchio [Rocchio 71] who defined an optimum retrieval function based on the cosine similarity function. However, Rocchio could not give a theoretical proof for his approach.

Wong et al. formulate an "acceptable ranking strategy" for optimum retrieval. They assume that a user gives "preference relations" for document pairs (d, d') instead of relevance judgements for single documents (The concept of preference relations is more powerful than that of relevance scales, but for reasons of simplicity we will restrict the following discussion to a binary relevance scale). The fact that a user prefers document d' over d is denoted as $d \prec\bullet d'$.

Now linear retrieval functions of the form $\vec{q}^T \cdot \vec{d}$ are regarded, where $\vec{q}$ represents the current request and $\vec{d}$ a document. Then the acceptable ranking strategy can be formulated: For any two pairs d, d' the following implication should hold:

$$d \prec\bullet d' \Longrightarrow \vec{q}^T \cdot \vec{d} < \vec{q}^T \cdot \vec{d}'.$$

This means that for any document pair for which the user has given a preference relation, the retrieval function should yield the correct ranking.

For the task of finding a vector $\vec{q}$ which fulfills the above criterion for a given sample of documents with preference relations, Wong et al. propose a gradient descent procedure. This procedure was originally developed for pattern recognition problems [Duda & Hart 73, pp. 138–147].

In our view, the major weakness of the approach described above is that it is well suited for retrospective retrieval where complete relevance information is available, but of limited value for the predictive case. Duda & Hart already stated "Of course, even if a separating vector is found for the design samples, it does not follow that the resulting classifier will perform well on independent test data" [Duda & Hart 73, p. 150].

A second problem with the "acceptable ranking strategy" occurs in the case when no optimum solution exists (e.g. when two documents with the same representation have different relevant judgements), because the gradient descent procedure does not converge in this case. For this problem, Duda & Hart proposed minimum squared error procedures, and these procedures are in fact preliminary versions of the LSP approach.

5 Experimental setting

For the experiments described in the following section, we took the collection from the AIR retrieval test [Fuhr & Knorz 84] with the (Boolean) search formulations relating to controlled language terms (called descriptors here) only. First, a very broad Boolean search with binary indexing was performed. For that, we applied a small cut-off value of 0.01 to the probabilistic indexing called A1 in [Fuhr & Knorz 84] which is based on a LSP indexing

function. In the following, only the sets of output documents selected this way are regarded. We compare the effect of different retrieval functions on the ranking of the documents within the output sets.

The 244 queries from the AIR retrieval test that had non-empty answer sets for the Boolean search were divided randomly into three samples named A, B and C. For the development of the LSP retrieval functions, we used sample B as learning sample and samples A and C as test samples. The number of queries and request-document pairs in each sample is shown in table 1. The multi-valued scale that has been used for the relevance judgements and the distribution of these judgements in the output sets is depicted in table 2.

Most of the experiments described in the following sections (unless stated otherwise) were performed using a binary relevance scale, where all documents not judged "irrelevant" or "digressive" were treated as being relevant.

For evaluation, we use the normalized recall measure as defined in [Bollmann et al. 86] for multi-valued relevance scales. This measure only considers documents in different ranks and with different relevance judgements. A pair of these documents is in the right order if the document with the higher relevance judgement comes first, otherwise it is in the wrong order. Let S^+ be the number of document pairs in the right order, S^- the number of those in the wrong order, and S^+_{max} the number of documents in the right order for an optimum ranking. The normalized recall is then defined as follows:

$$R_{norm} = \frac{1}{2}\left(1 + \frac{S^+ - S^-}{S^+_{max}}\right)$$

A random ordering of documents will have a R_{norm} value of 0.5 in the average. For the cases with S^+_{max}=0 we defined R_{norm}=1. Because of the large scattering of the answer sizes, we use two average methods besides the macro average (arithmetic mean) R^M_{norm}:

- the micro-macro average R^m_{norm} is a weighted average with respect to the answer sizes. Let n_i be the answer size of retrieval result Δ_i, then the micro-macro average of R_{norm} for a set of t queries is defined as:

$$R^m_{norm}(\Delta_1, \ldots, \Delta_t) = \frac{\sum_{i=1}^{t} n_i \cdot R_{norm}(\Delta_i)}{\sum_{i=1}^{t} n_i}$$

- the micro average R^μ_{norm} is computed by first combining the rank orders of different queries and then computing the R_{norm} measure for this combined rank order. There are several methods of combining rank orders for the computation of micro measures [Keen 71]. Here we use the retrieval status value computed by the retrieval function as criterion. This way, the R^μ_{norm} measure takes into account the query-wise ranking of the documents as well as the query-independent interpretation of the retrieval status values assigned by the retrieval function. This property is important for the evaluation of retrieval functions that are assumed to compute estimates of the probability of relevance, because these probabilities also have a query-independent interpretation.

6 Experimental results

We have performed numerous experiments with LSP retrieval functions ([Fuhr 88], [Konstantin 85]). Because of the limited space, we only desribe the major results here.

In order to develop LSP retrieval functions, first a description vector has to be defined. This description vector is derived from the representations of requests and documents. In our case, requests are represented by a set of descriptors (the Boolean connectives were dropped here), and documents by a set of pairs (descriptor, indexing weight). The elements of the corresponding description vector $\vec{x}(f_l, d_m)$ are listed in table 3.

The following discussion refers to two polynomials derived from the description vector listed in table 3:

$$\begin{aligned} v_1(\vec{x}) &= a_0 + a_1x_1 + a_2x_3 + a_3x_4 + a_4x_5 + a_5x_6 \\ &\quad + a_6x_7 + a_7x_9 + a_8x_{10} + a_9x_{11}x_{12} \\ &\quad + a_{11}x_5^2 + a_{12}x_5x_6 + a_{13}x_5x_{10} \\ &\quad + a_{14}x_6^2 + a_{15}x_6x_{10} + a_{16}x_{10}^2 \\ v_2(\vec{x}) &= a_0 + a_1x_1 + a_2x_2 + a_3x_8 + a_4x_9 + a_5x_{10} \\ &\quad + a_6x_{11} + a_7x_{12} + a_8x_1^2 + a_9x_1x_8 \\ &\quad + a_{10}x_1x_{10} + a_{11}x_8^2 + a_{12}x_8x_{10} \\ &\quad + a_{13}x_{10}^2 \end{aligned}$$

As can be seen from table 3, the elements $x_9 \ldots x_{13}$ will be constant for one request, so their value is not needed for a pure request-specific ranking of the documents. However, the LSP approach needs this information for the estimation of the probability of relevance $P(R|\vec{x}(f_l, d_m))$. Experiments without these elements showed inferior results with respect to all three measures R^μ_{norm}, R^M_{norm} and R^m_{norm}. Because of these results, we regarded a variation v_1', v_2' of the polynomials listed above where each occurrence of one of the elements x_{10}, x_{11}, x_{12} is replaced by the element x_{13}: this element gives the probability of relevance of an arbitrary document from the output set. The idea behind this was that we wanted to ease the adaption process of the coefficients: As the LSP approach attempts

to produce good probability estimates, the retrieval results are optimized with respect to R^{μ}_{norm}. With the additional information about the average probability of relevance for all documents of a query, this task becomes easier. Now the hope was that, (as a side effect) the request-oriented ranking (measured by R^{M}_{norm} and R^{m}_{norm}) also is improved. It is obvious that the value of x_{13} would not be available in a real application, but this would only affect the estimation of the probability and thus the results of R^{μ}_{norm} (marked by an asterisk here).

The results for the four polynomials are shown in table 4. The R^{μ}_{norm} values follow the theoretical considerations described above: with the element x_{13} instead of x_{10}, x_{11}, x_{12}, we get better results. However, the R^{m}_{norm} and R^{M}_{norm} values show that the request-specific ranking gets worse — just the contrary of what we expected. Other experiments where the polynomials v_1 and v_2 were extended by the element x_{13} showed similar results. So we end up with the conclusion that LSP retrieval functions are optimized with respect to $P(R|\vec{x}(f_l, d_m))$ and thus yield good R^{μ}_{norm} values, but this does not necessarily imply a good request-specific ranking of the documents.

For comparison, table 4 also shows the results for the cosine similarity function. This retrieval function yields results similar to those of the LSP approach. We think that there are two reasons why the LSP approach does not perform better in this application:

- The representation of requests and documents are simple and well suited to the application of the cosine measure. Especially, the probabilistic indexing weights of the documents cover all the information necessary for achieving a good ranking (this also has been shown for probabilistic models, see [Fuhr 86], [Fuhr 89]). The main strength of the LSP approach is its ability to handle complex representations, e.g. in the application for the development of indexing functions ([Fuhr & Knorz 84], [Biebricher et al. 88]). In fact, the indexing weights used for our ranking experiments were computed by application of LSP indexing functions.

- Our learning sample is too small with respect to the number of requests. As there are 82 queries in sample B, this means that we have only 82 (statistically) independent elements from which we derive the request-specific properties x_{10} – x_{13}. The experiments described above have shown the importance of request-specific information for the adaption process of the LSP retrieval functions. So we assume that we would perform better with a larger number of requests in the learning sample.

In all experiments described above, a binary relevance scale was used for the adaption of the coefficient vectors $\vec{a}$. In order to investigate the influence of the choice of the relevance scale on the final retrieval quality, experiments with a multi-valued scale were performed. Therefore we defined the following cost factors for the retrieval of a document with relevance judgement C_k : C_0=C_1=1, C_2=0.7, C_3=0.5, C_4=0.3, C_5=0. The results achieved with this multi-valued scale were almost the same as with the binary scale. So it seems that non-binary relevance scales do not give more information (in a stochastic sense) than a binary scale. Therefore, the restriction of former work on probabilistic models with binary relevance scales does not have any influence on the findings that were achieved.

7 Conclusions

In this paper, we have discussed the probabilistic concept of optimum retrieval. We have shown that actual retrieval functions cannot reach this optimum, because it is impossible to estimate the correct probabilities for each specific representation of documents or requests. Therefore, additional simplifying assumptions are necessary. For most probabilistic models, certain independence assumptions are postulated, so the performance of a model depends on the extent to which the assumptions fit to reality. With the LSP approach, the situation is quite different: here a class of polynomial retrieval functions is defined on the basis of some heuristics, and then the function that fits best to the available data is selected. Of course, the definition of the description vector and the class of polynomials is also based on some implicit assumptions. However, these assumptions are in general more vague, and it is not required to state them as explicitly as with the other probabilistic models.

The crucial point with this approach is the selection of features that make up the description vector. It would be desirable to have some theoretically well-founded method for this step. Another weakness of the LSP approach is the need of large learning samples, which is due to the problem of parameter estimation (see [Fuhr & Hüther 88], [Fuhr & Hüther 89]). Therefore, appropriate estimation methods for small samples should be developed, e.g. by modifying the learning data. Finally, the LSP approach is not suited to log-linear functions that are suggested by some theoretical models.

In contrast to most other types of retrieval functions, the LSP approach yields estimates of the pro-

bability of relevance and can also cope with multi-valued relevance scales. The major advantage of the LSP approach is its ability to deal with complex representations of retrieval objects. Especially for knowledge-based retrieval methods (see e.g. [Croft 87]), this approach is well suited: The representation is more complex then in the simple term-based case, and some kind of weighting scheme will be needed in any case. For this problem LSP functions can yield an optimum weighting based on the Probability Ranking Principle.

In the areas of pattern recognition and of machine learning, a number of sophisticated procedures for classifying complex objects have been developed. These methods should be considered with respect to their applicability in the field of information retrieval, especially those that are based on a probabilistic model: they have a well-founded theoretical background and can be shown to be optimum with respect to certain reasonable restrictions.

References

Biebricher, P.; Fuhr, N.; Knorz, G.; Lustig, G.; Schwantner, M. (1988). The Automatic Indexing System AIR/PHYS — from Research to Application. In: Chiaramella, Y. (ed.): *11th International Conference on Research and Development in Information Retrieval*, pp. 333–342. Presses Universitaires de Grenoble, Grenoble, France.

Bollmann, P.; Jochum, R.; Reiner, U.; Weissmann, V.; Zuse, H. (1986). Planung und Durchführung der Retrievaltests. In: Schneider, H. e. a. (ed.): *Leistungsbewertung von Information Retrieval Verfahren (LIVE)*, pp. 183–212. TU Berlin, Fachbereich Informatik, Computergestützte Informationssysteme (CIS), Institut für angewandte Informatik.

Bookstein, A. (1983). Outline of a General Probabilistic Retrieval Model. *Journal of Documentation 39(2)*, pp. 63–72.

Bookstein, A. (1988). *Loglinear Analysis of Library Data.* Research Report, OCLC, Office of Research.

Croft, W. B. (1987). Approaches to Intelligent Information Retrieval. *Information Processing and Management 23(4)*, pp. 249–254.

Duda, R. O.; Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* Wiley, New York.

Fuhr, N.; Hüther, H. (1988). Optimum Probability Estimation Based on Expectations. In: Chiaramella, Y. (ed.): *11th International Conference on Research and Development in Information Retrieval*, pp. 257–273. Presses Universitaires de Grenoble, Grenoble, France.

Fuhr, N.; Hüther, H. (1989). Optimum Probability Estimation from Empirical Distributions. To appear in: *Information Processing and Management 25.*

Fuhr, N.; Knorz, G. (1984). Retrieval Test Evaluation of a Rule Based Automatic Indexing (AIR/PHYS). In: Van Rijsbergen, C. J. (ed.): *Research and Development in Information Retrieval*, pp. 391–408. Cambridge University Press, Cambridge.

Fuhr, N. (1986). Two models of retrieval with probabilistic indexing. In: Rabitti, F. (ed.): *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval*, pp. 249–257. ACM, New York.

Fuhr, N. (1988). *Probabilistisches Indexing und Retrieval.* Fachinformationszentrum Karlsruhe, Eggenstein-Leopoldshafen.

Fuhr, N. (1989). Models for Retrieval with Probabilistic Indexing. *Information Processing and Management 25(1)*, pp. 55–72.

Keen, E. M. (1971). Evaluation Parameters. In: Salton, G. (ed.): *The SMART Retrieval System — Experiments in Automatic Document Processing*, pp. 74–112. Prentice Hall, Englewood Cliffs, New Jersey.

Knorz, G. (1983a). *Automatisches Indexieren als Erkennen abstrakter Objekte.* Niemeyer, Tübingen.

Knorz, G. (1983b). A Decision Theory Approach to Optimal Automatic Indexing. In: Salton, G.; Schneider, H. (ed.): *Research and Development in Information Retrieval*, pp. 174–193. Springer, Berlin et al.

Konstantin, J. (1985). *Untersuchung von nach dem Quadratmittel-Polynomansatz erstellten Rankingfunktionen.* Diplomarbeit, TH Darmstadt, FB Informatik, Datenverwaltungssysteme II.

Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworths, London, 2nd edition.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation 33*, pp. 294–304.

Robertson, S. E.; Maron, M. E.; Cooper, W. S. (1982). Probability of relevance: A unification of two competing models for document retrieval. *Information Technology: Research and Development 1*, pp. 1–21.

Rocchio, J. J. (1971). Relevance Feedback in information retrieval. In: Salton, G. (ed.): *The SMART Retrieval System — Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, New Jersey.

Salton, G. (ed.) (1971). *The SMART Retrieval System — Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, New Jersey.

Schürmann, J. (1977). *Polynomklassifikatoren für die Zeichenerkennung. Ansatz, Adaption, Anwendung.* Oldenbourg, München, Wien.

Wong, S. K. M.; Yao, Y. Y.; Bollmann, P. (1988). Linear Structure in Information Retrieval. In: Chiaramella, Y. (ed.): *11th International Conference on Research & Development in Information Retrieval*, pp. 219–232. Presses Universitaires de Grenoble, Grenoble, France.

Appendix

	total	A	B	C
# requests	244	79	82	83
# req.-doc. pairs	8476	2835	2822	2819

Table 1: Number of requests and request-document pairs in the different samples

retr.fct.	sample	R^{μ}_{norm}	R^{M}_{norm}	R^{m}_{norm}
v_1	B	0.752	0.754	0.717
v'_1		0.774*	0.751	0.716
v_2		0.752	0.763	0.717
v'_2		0.771*	0.745	0.714
v_1	C	0.721	0.753	0.717
v'_1		0.771*	0.740	0.708
v_2		0.721	0.714	0.700
v'_2		0.769*	0.710	0.699
cosine		0.668	0.728	0.704
v_1	A	0.741	0.775	0.723
v'_1		0.764*	0.756	0.715
v_2		0.741	0.771	0.731
v'_2		0.778*	0.760	0.729
cosine		0.727	0.769	0.731

Table 4: Results of the polynomial retrieval functions and the cosine measure

judgement		# pairs with judgement		sample A	sample B	sample C
relevant	r_5	2258	(27%)	732	776	750
conditionally relevant/more relevant	r_4	729	(9%)	200	217	312
conditionally relevant	r_3	440	(5%)	135	160	145
conditionally relevant/more irrelevant	r_2	716	(8%)	249	241	216
irrelevant	r_1	3489	(41%)	1204	1057	1228
digressive	r_0	844	(10%)	315	371	158
total		8476	(100%)	2835	2822	2819

Table 2: Distribution of relevance judgements in the output sets

element	description
x_1	# descriptors common to query and document
x_2	log(# descriptors common to query and document)
x_3	highest indexing weight of a common descriptor
x_4	lowest indexing weight of a common descriptor
x_5	# common descriptors with weight ≥ 0.15
x_6	# non-common descriptors with weight ≥ 0.15
x_7	# descriptors in the document with weight ≥ 0.15
x_8	log$\sum$(indexing weights of common descriptors)
x_9	log(# descriptors in the query)
x_{10}	log(min(size of output set, 100))
x_{11}	= 1, if size of output set > 100
x_{12}	= 1, if request about nuclear physics
x_{13}	proportion of relevant documents in the output set

Table 3: Elements of the description vector $\vec{x}(f_l, d_m)$