Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

ELSEVIER International Journal of Approximate Reasoning 34 (2003) 265–285
www.elsevier.com/locate/ijar

The BNR model: foundations and performance of a Bayesian network-based retrieval model

Luis M. de Campos *, Juan M. Fernández-Luna, Juan F. Huete

Departamento de Ciencias de la Computación, e Inteligencia Artificial, E.T.S.I. Informática, Universidad de Granada, 38, 18071 Granada, Spain

Received 1 March 2003; accepted 1 July 2003

Abstract

This paper presents an information retrieval model based on the Bayesian network formalism. The topology of the network (representing the dependence relationships between terms and documents) as well as the quantitative knowledge (the probabilities encoding the strength of these relationships) will be mined from the document collection using automatic learning algorithms. The relevance of a document to a given query is obtained by means of an inference process through a complex network of dependences. A new inference technique, called propagation + evaluation, has been developed in order to obtain the exact probabilities of relevance in the whole network efficiently.
© 2003 Elsevier Inc. All rights reserved.

1. Introduction

Nowadays, the retrieval of information is becoming more and more important with the widespread use of Internet in our everyday tasks. The field of information retrieval (IR) has been defined by Salton and McGill [22] as the

* Corresponding author. Tel.: +34-58-443-199/958-244019; fax: +34-58-243-317/958-243317.
E-mail addresses: lci@decsai.ugr.es (L.M. de Campos), jmfluna@decsai.ugr.es (J.M. Fernández-Luna), jhg@decsai.ugr.es (J.F. Huete).

0888-613X/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2003.07.011

subject concerned with the representation, storage, organization, and accessing of information items. [1] In this paper, we mainly focus our attention on two main IR tasks: representing the information, and the way in which we access information items, i.e. identifying documents in a collection that are relevant to a particular information need formulated by means of a query.

We shall focus our research on the use of uncertain inference models for IR [3]. These models represent an extension of the classical probabilistic model [28], providing a framework for the integration of several sources of evidence. The use of these models is based on the fact that most tasks in this area may be described as uncertain processes [26]. The theoretical justification for these models is based on the 'probability ranking principle' [21] which states that the best overall retrieval effectiveness will be achieved when documents are ranked in decreasing order of their probability of relevance.

The concept of relevance in uncertain inference models is basically related to an inference process through a network of dependences using evidential reasoning techniques. The most promising ones are those based on Bayesian networks [18]. Intuitively, as [16] says, "Bayesian networks are complex diagrams that organize the body of knowledge in a given area by mapping out cause-and-effect relationships among key variables and encoding them with numbers that represent the extent to which one variable is likely to affect another". The use of a general Bayesian network methodology as the basis for an IR system is difficult to tackle. The problem mainly appears because of the large number of variables involved and the computational efforts needed to both determine the relationships between variables and perform the inference processes. [2] Nevertheless, an increasing effort has been made in the research of uncertain inference models for IR [17,20,25]. These models consider the following two main simplifying restrictions in order to solve the above efficiency problem:

R1 Fixed dependence relationships: the structure of the model, encoding the dependence relationships between variables, is fixed a priori, without considering any potential knowledge that might be mined from the collection.

R2 Simplified estimation of probabilities: in order to avoid the large space necessary to store all the probabilities relevant to the process, it is assumed that those complex compound events will have been assigned zero probability values. With this assignment, these events can be discarded when inference tasks are performed.

Using the restrictions above, the probability of relevance of a given document only depends on the set of terms used to formulate the query and it can be

[1] In this paper, we will only deal with documents, or in a broader sense, textual representations of any type of object, i.e. a research article, a book, a message in an electronic mail file, etc.

[2] Note that these tasks are NP-hard [10] in the number of variables.

computed without truly performing inference tasks, i.e. without propagating the evidences through the networks. In these cases, it is sufficient to consider the evaluation of a set of functions.

Our objective in this paper is to show that it is possible to relax these restrictions, and therefore to obtain a more expressive Bayesian network-based IR model. This relaxation involves considering new theoretical and practical trends: how to infer the set of relationships between variables from the collection, how to estimate and store all the needed probabilities efficiently, and finally, assuming that we have the solutions for the previous problems, it will be necessary to study how to perform exact inference efficiently through the network.

Following these ideas, this paper is divided into the following sections: in Section 2, we introduce the Bayesian network background needed to understand the rest of the paper. Section 3 presents other models based on these graphical models. Section 4 will explain the Bayesian Network Retrieval Model (BNRM) in detail: its topology and construction, the estimation of probability distributions, and the inference method. In Section 5, the results of an experimentation with this new model is presented. The performance of the model is also compared with the effectiveness of other models such as the vector space model and inference network model. Finally, Section 6 shows the conclusions of this work, as well as future work that we plan to implement in order to improve the BNRM.

2. Preliminaries: Bayesian networks basics

In this section, we shall briefly introduce the concept of the Bayesian network [18], the basis for the model presented in this paper. We shall attempt to answer questions such as what it is for, how it is composed, how it can be constructed, and how it can be used.

In formal terms, a Bayesian network is a directed acyclic graph (DAG) (a graph with links which are orientated, taking the name of arcs, and with no cycles in it), in which the nodes represent random variables and the arcs show causality, relevance or dependency relationships between them. [3] The variables and their relationships comprise the qualitative knowledge stored in a Bayesian network. A second type of knowledge also stored in the DAG is known as quantitative, since it establishes the strength of the relationships and is measured by means of probability distributions. Associated with each node there is

[3] A dependence relationship between two variables, X and Y, implies a modification of the belief in X, given that the value taken by Y is known. An Independence relationship means that the belief in X is not modified, given the knowledge on Y.

a set of conditional probability distributions, one for each possible combination of values that its parents can take.

Formally, a Bayesian network can be considered an efficient representation of a joint probability distribution that takes into account the set of independence relationships represented in the graphical component of the model. In general terms, given a set of variables $\{X_1, \ldots, X_n\}$ and a Bayesian network $\mathscr{G}$, the joint probability distribution in terms of local conditional probabilities is obtained as follows

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \pi(X_i))$$

where $\pi(X_i)$ is any combination of the values of the parent set of X_i, $\Pi(X_i)$, in the graph. If X_i has no parents, then the set $\Pi(X_i)$ is empty, and therefore $P(X_i|\pi(X_i))$ is just $P(X_i)$.

Once completed, a Bayesian network can be used to derive the posterior probability distribution of one or more variables since we have observed the particular values for other variables in the network, or to update previous conclusions when new evidence reach the system. Researchers have developed general inference algorithms that take advantage of the independences represented in the network. Although it is possible to find algorithms that perform inference tasks in a time that is linear in the number of variables, high computational complexity inference algorithms result from having multiple pathways connecting nodes in the graph. General inference has been proved to be NP-hard [10].

3. Related Bayesian network-based models

In this section we shall briefly describe the two main retrieval models based on Bayesian networks.

The first model was developed by Croft and Turtle [25], the *Inference Network Model*, which is composed, in its simplified form, of two networks: the document and query networks. The first, fixed for a given collection, represents the document collection, and contains two kinds of nodes: the document nodes, representing the documents, and the concept nodes, symbolizing the index terms contained in the documents. The arcs go from each document node to each concept node used to index it. In addition, the query network is specific for each query. In the simplified form, there is a query node for each query representation used to express the information needed. This query node has as parents those concepts (terms) used to formulate the query, representing the connection between the two networks. The query nodes are also the parents of an information need node that represents the user's generic information need.

When there is only one query representation the information need node and the query node coincide. In the rest of the paper, we shall concentrate our attention on those models which use only one query representation. Fig. 1, on the left hand side, shows this simplified Inference network. In order to complete it, it is necessary to assess the conditional probability distributions of the nodes in the graph. The proper specification of these probabilities allows the inference network to cover different IR strategies.

A document D_j may be ranked with respect to a query Q by measuring how much evidential support the observation of D_j provides to the query Q. In order to obtain this ranking, a single document node D_j is instantiated each time, and the probability that the information need is satisfied given that this document has been observed, $p(Q \wedge D_j = \text{true})$, is computed.

A direct computation of these values is unfeasible for practical purposes. In order to solve this problem, the inference network takes advantage of a particular probability assessment. It is interesting to note that using these probabilities and considering that in the inference process there is only one document instantiated to relevant, the final probability $p(Q \wedge D_j = \text{true})$ only depends on the set of terms indexing document D_j that have been used to formulate the query.

We shall now present a second model based on BNs: the Belief Network Model [20]. This model has been designed to provide a Bayesian network-based approach capable of simulating the vector space, and Boolean and probabilistic schemes. Like the inference network model, their network is composed of three types of nodes: document nodes, concept (term) nodes, and the query node. The arcs go from concept nodes to the document nodes where they occur, and from the concept nodes (appearing in the query) to the query node. This model is represented on the right hand side of Fig. 1. The ranking will be obtained by computing the probability $p(D_j = \text{relevant}|Q)$ for each document D_j.

Since a document can be indexed by hundred of terms, a straight computation of this probability becomes unfeasible. Therefore, and like the inference

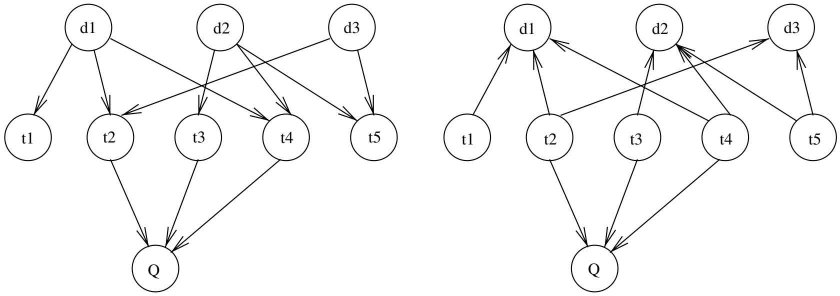

Fig. 1. Inference network and belief network model.

network model, the probabilities are defined in such a way that the computation will be reduced to a direct evaluation of a function. Also, depending on the model to be simulated, different probability assessments might be used. Using the probability assessment, in order to compute the final ranking of a document, the individual contribution of each term of the document which also belongs to the query is considered.

In short, both models use a fixed document subnetwork structure for a given collection and a degenerated probability assessment in order to compute the probabilities of interest without truly propagating the evidences through the networks. Since these models use the same dependence model to represent any document collection, they do not take into account the particular dependence relationships between variables (terms and/or documents) that can be mined from the document collection. In addition, the computed probabilities of relevance will only depend on the terms used to formulate the query and not upon other concepts that might be related (either directly or indirectly) to the query. In the following section, we shall present our model to try to reduce these problems.

4. Foundations of the Bayesian network retrieval model

Our objective is to obtain a Bayesian network representation of a given document collection. Particularizing the definition given in [16] to the field of IR, a Bayesian networks will organize the knowledge that can be mined from a document collection by mapping out dependence relationships between terms and documents and encoding them with probabilities representing the extent to which they are related to each other. Once the network has been constructed, it shall be used to obtain the relevant documents for a given query.

With this definition in mind, in our approach we shall not include a query component (query nodes) as a proper part of the IR system, i.e. it will be a query independent model. In our case, the query is considered as an evidence that should be introduced into the system. This fact shall represent the first difference between our model and previous ones.

In order to present the BNRM, we shall first describe how we can determine the dependence relationships, i.e. the qualitative component; then we will present the assessment of the probability values, that is to say, the quantitative component; and finally, we shall consider how the inference process is carried out.

4.1. Structure of the model

Since our objective is to obtain a model able to incorporate the most important dependence relationships in the collection, a learning procedure must be performed. This process will give the final structure of the network as a result. In our model, we shall consider two sets of variables: *Terms* ($\mathscr{T} = \{T_i,\ i = 1, \ldots, M\}$, with M being the number of terms used to index the collection) and *Documents* ($\mathscr{D} = \{D_j,\ j = 1, \ldots, N\}$, N being the total number of documents). These variables are bivaluated, taking values from the set $\{$*relevant, not-relevant*$\}$.[4] In order to simplify the following expressions, we shall note the 'term T_i (or D_j for documents) is relevant' as $t_i(d_j)$, and 'the term $T_i(D_j)$ is not relevant' as $\bar{t}_i(\bar{d}_j)$.

Given a document collection, and due to the large number of variables involved, mining all the dependence relationships is unfeasible. We propose that a hybrid approach be followed whereby both "expert knowledge" (using a set of general coherence criteria) and "collection knowledge" (mined from the documentary data) will be taken into account.

4.1.1. Expert knowledge

Consists of a set of assumptions that the model must fulfill and allows us to obtain a previous skeleton of the Bayesian network (limiting the automatic learning process). These assumptions are

1. There is a strong relationship between a document and each of the terms by which it has been indexed. This principle is translated into the graph using links that connect each document node with all the term nodes that represent the index terms associated to the corresponding document.
2. The relationships between documents are only present through the terms that index them. This assumption implies that there are no links joining the document nodes between them.
3. If we know the relevance or non-relevance of all the terms that occur in a document, D_i, our belief in its relevance is not affected by the fact that we know that another document, D_j, or term, T_k, are relevant or not. This assumption implies that documents are conditionally independent given the terms by which they have been indexed. In the network, the links joining the document nodes and their corresponding term nodes will be directed from the second to the first ones.

Taking these three assumptions into account, the structure of our model is similar to the Belief Network Model [20], except for the fact that we do not consider a query node. In this network, the terms are independent between each other. This point seems to be very restrictive because, in a collection, the

[4] We talk about the relevance of a term in the sense that the user believes that the term will appear in relevant documents (hence, he/she will explicitly use it when formulating a query). Similarly, a term is not relevant when users believe that the relevant documents do not contain it: they are not interested in documents containing this term.

terms are related in different ways. This restriction will be removed by considering collection-dependent knowledge.

4.1.2. Collection knowledge

Considering the above assumptions, the natural step for obtaining a more precise model is to incorporate the most important dependence relationships between the terms into the collection. Thus, we may distinguish two different layers of nodes: the term and the document layer. As we will see in Section 4.3, the separation in these two layers will allow inferences to be carried out efficiently.

In order to put this methodology into practice, we must use an automatic learning algorithm to build the term layer. The first task is to decide the underlying topology of the term layer. It is obvious that the more complex the topology, the more accurate the dependence and independence relationships will be reflected by the topology, although, at the same time, and considering the number of terms and documents involved, the learning and propagation algorithms will be a very time-consuming tasks. In order to tackle this problem, we propose that simpler graphs be used. In this case, some precision is lost because the independence and dependence relationships that they can represent are more restrictive.

In this paper we consider that term to term dependences will be represented by means of a *polytree* [5] because there is a set of very efficient learning [4,5,19] and propagation [18] algorithms running in a time proportional to the number of nodes, making the use of Bayesian networks in this context feasible. In particular, the term layer will be completed using a polytree learning algorithm which takes as input the inverted file of a collection (a data structured that stores for each term those documents where it occurs), and generates a polytree, whose nodes (variables) are the terms.

The algorithms, which is explained in detail in [5], is composed of three main steps:

1. Computation of the degrees of dependency between all pairs of nodes.
2. Construction of the tree skeleton.
3. Orientation of the edges in the tree, finally making up a polytree.

Several remarks have to be made about these three parts. First, the measure used to establish the dependency between nodes (which is, in some sense, analogous to the functions usually employed in IR systems for measuring the similarity between the terms in the collection) is the following:

$$\mathrm{Dep}(T_i, T_j|\emptyset) = \sum_{\mathbf{T_i},\mathbf{T_j}} p(\mathbf{T_i}, \mathbf{T_j}) \ln\left(\frac{p(\mathbf{T_i}, \mathbf{T_j})}{p(\mathbf{T_i})p(\mathbf{T_j})}\right) \tag{1}$$

[5] Graph in which there is no more than one directed path connecting each pair of nodes.

where $\mathbf{T_i}$ is one of the possible values that the variable T_i can have. This function is the Kullback–Leibler's cross entropy (also called expected mutual information measure), which measures the dependency degree between two variables T_i and T_j (which is equal to zero if T_i and T_j are marginally independent, and such that the more dependent T_i and T_j are, the greater $\mathrm{Dep}(T_i, T_j|\emptyset)$ is). The probabilities $p(\mathbf{T_i}, \mathbf{T_j})$ are estimated from the inverted file by counting frequencies. Here, we use the marginal cross entropy, in opposition to the approach in which the marginal dependency of two terms is combined with the conditional dependencies of these two terms conditioned to the rest of terms. The reason is that due to the great amount of terms in a collection, the computation of the conditional dependencies, although it has to be carried out only once, has been proved extremely time-consuming but also a large storage is needed.

The next step is the tree skeleton construction. If we assume that the computed dependency values are link weights in a graph, this algorithm gets a maximum weight spanning tree (MWST), i.e. a tree where the sum of the weights of its links is maximum. We considered the Prim's algorithm [2] to obtain the MWST.

Due to the great number of terms that there are generally in a collection, the values of the dependencies are very low in general, and sometimes the algorithm does not have any good choice and selects as the highest value among all the dependencies being considered a very low value, adding the corresponding link to the tree. The problem lies in the fact that the two linked nodes are almost more independent than dependent, and therefore the model we are building loses accuracy with respect to the original one.

To solve this problem, the algorithm, once it has selected a new link T_i–T_j to be added to the tree, performs an independency test between T_i and T_j; then it really adds this link to the tree only if the independency test fails. In this way, we can obtain a non-connected tree, i.e., a forest, as the result of this step.

Once the skeleton is built, the last part of the learning algorithm deals with the orientation of the tree, getting as a result a polytree. In a head to head pattern $T_i \rightarrow T_k \leftarrow T_j$, the instantiation of the head to head node T_k should normally increase the degree of dependency between T_i and T_j, whereas in a non-head to head pattern such as $T_i \leftarrow T_k \rightarrow T_j$, the instantiation of the middle node T_k should produce the opposite effect, decreasing the degree of dependency between T_i and T_j. So, we compare the degree of dependency between T_i and T_j after the instantiation of T_k, $\mathrm{Dep}(T_i, T_j|T_k)$, with the degree of dependency between T_i and T_j before the instantiation of T_k, $\mathrm{Dep}(T_i, T_j|\emptyset)$, and direct the edges toward T_k if the former is greater than the latter. Finally, the algorithm directs the remaining edges without introducing new head to head connections. This strategy produced, in our preliminary experiments, structures where several nodes had a great number of parents; this fact leads to have very big probability tables and, as a consequence, it causes problems of storage and reliability (in the estimation of these tables). For that reason we have restricted a bit the rule that produces head to head connections, by including another condition in the antecedent: we want to be sure that if we decide to include a head to head connection $T_i \rightarrow T_k \leftarrow T_j$, then the nodes T_i and T_j are not conditionally independent given T_i. So, we also test this condition, once again using a Chi square test of independency based on the value $\mathrm{Dep}(T_i, T_j | T_k)$ (in this case with two degrees of freedom).

Once the polytree has been learned, the last step to finish the retrieval model construction is to join each term node with its corresponding document node. Fig. 2 shows an example of the final topology of the network.

4.2. Estimating the quantitative information

Once the structure of the network has been created, the second step in specifying a Bayesian network completely is to estimate the strength of the relationships represented. This process implies estimating a set of conditional probability distributions. We have used several estimators [12], but the ones that perform best are the following.

4.2.1. Root term nodes

Given a root node representing the variable T_i, it will have to store the marginal probability of relevance, $p(t_i)$, and the probability of being non-relevant, $p(\bar{t}_i)$ defined by means of $p(t_i) = \frac{1}{M}$ and $p(\bar{t}_i) = 1 - p(t_i)$, with M being the number of terms in the collection.

4.2.2. Non-root term nodes

In this case, for each non-root term node T_i, with parents $\Pi(T_i)$, we need to estimate a set of conditional probability distributions $p(T_i|\pi(T_i))$, one for each possible combination of values that the parents of a node T_i can have, $\pi(T_i)$.

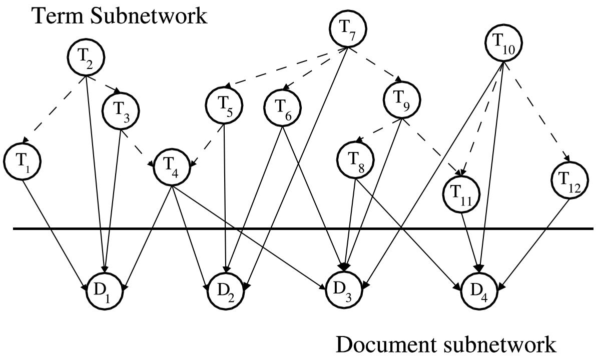

Fig. 2. The Bayesian network retrieval model.

Given any set of terms $\mathscr{S} = \{T_1, T_2, \ldots, T_k\}$, a configuration C is defined as a vector $\langle \mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_k} \rangle$, where each of its elements corresponds to a value that each variable $T_i \in \mathscr{S}$ can take. Therefore, $\mathbf{t_i} = t_i$ if the ith variable is relevant, and $\mathbf{t_i} = \bar{t}_i$, if T_i is not relevant. For instance, for $\mathscr{S} = \{T_1, T_2, T_3, T_4\}$, two possible configurations are $\langle t_1, t_2, \bar{t}_3, t_4 \rangle$ and $\langle t_1, \bar{t}_2, t_3, \bar{t}_4 \rangle$. Given a set of terms $\mathscr{S}$ and a configuration C, we define $n(C)$ as the number of documents that contains all the terms that are included as relevant in the configuration, and do not contain those that are non-relevant in it.

The estimator is based on the Jaccard similarity measure [28]. Given two sets X and Y, it computes the similarity between them as the quotient of the number of elements composing the intersection and the cardinality of the union of both sets, i.e. $|X \cap Y|/|X \cup Y|$. This measure (also used by Savoy [23]) is adapted to our model using the following expression:

$$p(\bar{t}_i|\pi(T_i)) = \frac{n(\langle \bar{t}_i, \pi(T_i) \rangle)}{n(\langle \bar{t}_i \rangle) + n(\pi(T_i)) - n(\langle \bar{t}_i, \pi(T_i) \rangle)} \tag{2}$$

In this formula, $p(\bar{t}_i|\pi(T_i))$ is initially estimated and later $p(t_i|\pi(T_i))$ is obtained by duality $(p(t_i|\pi(T_i)) = 1 - p(\bar{t}_i|\pi(T_i)))$.

4.2.3. Document nodes

In this case, the probability $p(D_j|\Pi(D_j))$ must be estimated, i.e. the probability of a document node given the set of its parents (the nodes representing the terms by which it has been indexed).

The main problem to be faced in this task is that if a document has been indexed by m_j terms, and taking into account that each term is represented by a binary variable, the number of probability distribution to be estimated is 2^{m_j}. Taking into account that in a common size collection, the number of index terms per document may be around 100 or 200, the total number of possible combinations is huge, leading to several problems such as the long time needed to estimate the probabilities, the low reliability of the estimation, the great amount of disk space required to store the distributions, and finally, the slowness of the propagation process to manage them. The existence of these four chained problems lead us to consider an alternative way to estimate the probability matrices completely, and resulted in what we have called *probability functions*, also known as *canonical models of multicausal interaction* [18].

In the inference process, the probability functions will compute the required conditional probabilities just at the moment when they were needed. In this way, the explicit representation of the probability matrix is substituted by an implicit one, avoiding most of the previously explained problems.

We have developed a new general canonical model: for any configuration $\pi(D_j)$ of $\Pi(D_j)$, we define the conditional probability of relevance of D_j as follows:

$$p(d_j|\pi(D_j)) = \sum_{T_i \in R_{\pi(D_j)}} w_{ij} \tag{3}$$

with $R_{\pi(D_j)}$ being the set of terms that are relevant in $\pi(D_j)$, and the weights w_{ij} have to verify that $0 \leqslant w_{ij}$ and $\sum_{T_i \in D_j} w_{ij} \leqslant 1$. So, the more relevant terms in $\pi(D_j)$, the greater the probability of relevance of D_j.

4.3. The retrieval engine: inference in the Bayesian Network Retrieval Model

Once the Bayesian network has been built, it can be used to predict the values that certain variables can take. This process is known as *inference* and computes the probabilities of the different cases that an unknown variable can have, given the values of the known variables or evidences.

Focusing on the BNRM, the query formulated by the user (or more specifically, the terms in the query) plays the role of a new piece of evidence provided to the system. The last aim is to obtain the probability of relevance of each document in the collection given a query. The terms from the query are instantiated to relevant in the network. This information will be propagated toward the document nodes, finally obtaining $p(d_j|Q), \forall D_j$. The documents are presented to the user decreasingly sorted according to their corresponding probabilities of relevance.

Taking into account the number of nodes in our Bayesian network and the fact that it contains cycles and nodes with a great number of parents, general purpose inference algorithms cannot be applied due to efficiency considerations, even for small document collections. Therefore, we ought to look for a solution to carry out the inference in an acceptable time. Our proposal for solving this problem has been named Propagation + Evaluation, and consists of a two-stage approximate propagation:

1. Exact propagation in the term layer, obtaining $p(t_i|Q), \forall T_i$. Bearing in mind that the evidences will always be term nodes composing the query, we could use Pearl's exact propagation algorithm [18] in order to obtain the posterior probability of each term node. These probabilities can be computed in a polynomial time in an exact way.
2. Evaluation of a probability function in the document nodes, computing $p(d_j|Q)\ \forall D_j$, using the posterior probabilities obtained in the previous stage. With this evaluation, we are modifying the strength with which the terms influence the relevance of the documents.

Therefore, the computation of $p(d_j|Q)$ can be carried out as follows:

$$p(d_j|Q) = \sum_{i=1}^{m_j} w_{ij} \cdot p(t_i|Q) \tag{4}$$

In the following theorem, we show the conditions under which we could put the two-stage propagation into practice, with total equivalence in results with respect to an exact propagation:

Theorem 4.1. *Given a set of evidences corresponding to the terms of a query Q, if the probability function used can be expressed as*

$$p(d_j|\pi(D_j)) = \sum_{T_i \in R_{\pi(D_j)}} w_{ij}, \quad \forall j = 1, \ldots, N \tag{5}$$

that is to say, as the sum of weights for the relevant terms of a document, where $0 \leqslant w_{ij}$, $\forall i = 1, \ldots, m_j$, $\sum_{T_i \in D_j} w_{ij} \leqslant 1$ *and* $R_{\pi(D_j)}$ *is the set of terms that are relevant in a configuration of parents of* D_j, $\pi(D_j)$, *then the exact propagation in the term layer plus the evaluation of a probability function in each document* (*Eq.* (4)) *is equivalent to carrying out an exact propagation in the entire Bayesian network.*

Proof. The posterior probability obtained applying to the exact inference process, $p(d_j|Q)$ can be expressed as

$$p(d_j|Q) = \sum_{\pi(D_j)} p(d_j|\pi(D_j), Q) \cdot p(\pi(D_j)|Q)$$

As the set of terms indexing a document makes the document and the evidences independent, then

$$p(d_j|Q) = \sum_{\pi(D_j)} p(d_j|\pi(D_j)) \cdot p(\pi(D_j)|Q)$$

Substituting in the previous expression the value of $p(d_j|\pi(D_j))$ in Eq. (5), we obtain:

$$p(d_j|Q) = \sum_{\pi(D_j)} \left(\sum_{T_i \in R_{\pi(D_j)}} w_{ij} \cdot p(\pi(D_j)|Q) \right) \tag{6}$$

The next step is to break down the previous expression into two parts. In the first, we include the configurations where the term T_{m_j} is relevant, and in the second, those where it is not relevant. In order to make this fact explicit, we will use notation $\langle \pi^*(D_j), \mathbf{t_{m_j}} \rangle$, where $(\pi^*(D_j))$ corresponds with the configuration $\langle \mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_{m_j-1}} \rangle$, i.e. without the last variable, T_{m_j}, in $\pi(D_j)$.

$$\begin{aligned} p(d_j|Q) = & \sum_{\langle \pi^*(D_j), t_{m_j} \rangle} \left(\sum_{T_i \in R_{\langle \pi^*(D_j), t_{m_j} \rangle}} w_{ij} \cdot p(\pi(D_j)|Q) \right) \\ & + \sum_{\langle \pi^*(D_j), \bar{t}_{m_j} \rangle} \left(\sum_{T_i \in R_{\langle \pi^*(D_j), \bar{t}_{m_j} \rangle}} w_{ij} \cdot p(\pi(D_j)|Q) \right) \end{aligned} \tag{7}$$

Considering that $\sum_{T_i\in R_{\langle\pi^*(D_j),t_{m_j}\rangle}} w_{ij}\cdot p(\pi(D_j)|Q)$ is equal to

$$\sum_{T_i\in R_{\langle\pi^*(D_j)\rangle}} w_{ij}\cdot p(\pi(D_j)|Q) + w_{m_jj}\cdot p(\pi(D_j)|Q)$$

and that $\sum_{T_i\in R_{\langle\pi^*(D_j),\bar{t}_{m_j}\rangle}} w_{ij}\cdot p(\pi(D_j)|Q) = \sum_{T_i\in R_{\langle\pi^*(D_j)\rangle}} w_{ij}\cdot p(\pi(D_j)|Q)$, and substituting them in expression (7), the posterior probability is

$$\begin{aligned} p(d_j|Q) = {} & \sum_{\langle\pi^*(D_j),t_{m_j}\rangle}\left[\sum_{T_i\in R_{\langle\pi^*(D_j)\rangle}} w_{ij}\cdot p(\pi(D_j)|Q) + w_{m_jj}\cdot p(\pi(D_j)|Q)\right] \\ & + \sum_{\langle\pi^*(D_j),\bar{t}_{m_j}\rangle}\left[\sum_{T_i\in R_{\langle\pi^*(D_j)\rangle}} w_{ij}\cdot p(\pi(D_j)|Q)\right] \end{aligned}$$

Notice how both sums in the configuration have $\sum_{T_i\in R_{\langle\pi^*(D_j)\rangle}} w_{ij}\cdot p(\pi(D_j)|Q)$ in common when T_{m_j} is taken into account and when it is disregarded. We could unify these two addends into only one by means of a marginalization operation over the variable T_{m_j}. Consequently, the variable T_{m_j} will be removed from the resultant addend.

Therefore, the posterior probability of the document will be

$$p(d_j|Q) = \sum_{\pi^*(D_j)}\sum_{T_i\in R_{\pi^*(D_j)}} w_{ij}\cdot p((\pi^*(D_j))|Q) + \sum_{\langle\pi^*(D_j),t_{m_j}\rangle} w_{m_jj}p(\pi(D_j)|Q)$$

Focusing our attention on the second addend in the previous expression:

$$\sum_{\langle\pi^*(D_j),t_{m_j}\rangle} w_{m_jj}p(\pi(D_j)|Q) = w_{m_jj}\cdot\sum_{\langle\pi^*(D_j),t_{m_j}\rangle} p(\pi(D_j)|Q)$$

which implies that we are considering all the possible configurations in $\pi^*(D_j)$, and therefore the final result is $w_{m_jj}\cdot p(t_{m_j}|Q)$. Note that $p(t_{m_j}|Q)$ has been obtained previously by applying the exact propagation process in the term layer.

Therefore,

$$p(d_j|Q) = \sum_{\pi^*(D_j)}\sum_{T_i\in R_{\pi^*(D_j)}} w_{ij}\cdot p((\pi^*(D_j))|Q) + w_{m_jj}\cdot p(t_{m_j}|Q)$$

It should be noted that the first addend is completely analogous to the initial expression, Eq. (6), but where the term T_{m_j} has been removed. We now repeat the process applied to this first addend to remove a new variable $T_{m_j}-1$ and extract the addend $w_{m_j-1j}\cdot p(t_{m_{j-1}}|Q)$. By repeating the process until we have removed all the terms, we obtain a final expression of the probability of a document given all the evidences:

$$p(d_j|Q) = \sum_{i=1}^{m_j} w_{ij}\cdot p(t_i|Q)$$

We conclude that we can compute the probability $p(d_j|Q)$ exactly, $\forall D_j$ running an exact propagation only in the term layer. □

4.4. Two modifications to the basic retrieval process in the BNRM

Until now, we have assumed that the presence of a term in a query implies its instantiation to relevant, assigning the same 'strength' to all of them. However, it might be interesting to highlight the importance of a term with respect to others that could be classified as secondary. This information is naturally used in the vector space model [22].

In the BNRM, these terms which occur more frequently in the query, will have a greater influence in the propagation stage than those that appear a few times. In order to put this idea into practice, we could clone these query terms in the network. Therefore, if the query frequency (qf) of a term T_i were three, then two new fictitious nodes would be created in the network with the same information contained in the node T_i. These nodes would be used in the evaluation of each document which was indexed by T_i. In this case, the posterior probability can be computed (see [12]) as

$$p(d_j|Q) = \sum_{i=1}^{m_j} w_{ij} \cdot p(t_i|Q) \cdot [\mathrm{qf}_i]$$

In this expression, the factor $[\mathrm{qf}_i]$ will have the value 1 if the ith term is not in the query, and the corresponding qf_i if it occurs, which is why it has been noted between brackets.

A second modification to the basic process set out in the previous subsection is the following: a high posterior probability of relevance after propagating might be due to a positive influence of the instantiated query terms on a document, or to the fact that the prior probability is high and the influence received by the query terms does not decrease the posterior probability of relevance of the document. The first case implies that the document is very relevant. However, the second means that if documents are ranked according to their posterior probabilities, mistakes can be made, and greater importance given to a document which has briefly increased its belief and therefore, worsening the retrieval performance.

Therefore, the ranking could be generated by taking the difference $p(d_j|Q) - p(d_j)$, $\forall D_j$ into account. In this case, the important fact is the relative value (the increment of probability) and not the particular value of $p(d_j|Q)$.

5. Measuring the performance of the model: experiments and results

Our objective in this section is to measure the effectiveness of the retrieval. This evaluation can be carried out with different methods, but the main one is

Table 1
Main features of the standard test collections

Collection	No. documents	No. terms	No. queries
ADI	82	828	35
CACM	3204	7562	64
CISI	1460	4985	112
CRANFIELD	1398	3857	225
MEDLARS	1033	7170	30

that based on *recall* and *precision* estimates [22,28]. The first measures the ability of the IR system to present all the relevant documents (recall = number of relevant documents retrieved/number of relevant documents). The second, precision, measures its ability to present only the relevant documents (precision = number of relevant documents retrieved/number of documents retrieved). For each experiment, we offer the mean precision for the eleven recall points [22], A-11PTS.

In this section, we shall present the experimentation that we have carried out in order to determine the quality of the proposed model. We have applied the BNRM to five well-known test document collections: ADI, CACM, CISI, CRANFIELD and MEDLARS. The main characteristics of these collections with respect to the number of documents, terms and queries are shown in Table 1.

For each collection, our objective is to show the behavior of the proposed methodology and to compare the obtained results with other IR systems like SMART, based on the vector space model, [6] and with other Bayesian network-based models such as the Inference Network Model.

All the collection has been indexed by SMART, not only for the experiments carried out by our system, but also for those run by SMART and INM. Specifically, after removing stop words, a stemming process is run, leading each word to its corresponding stem. The SMART weighting scheme with which the experiments have been run is "ntc", because it is the one with which this IRS obtains the best results with these five test collections.

The specific weights w_{ij}, for each document D_j and each term $T_i \in D_j$, used by our models (see Eq. (3)) are: [7]

$$w_{ij} = \alpha^{-1} \frac{\mathrm{tf}_{ij} \cdot \mathrm{idf}_i^2}{\sqrt{\sum_{T_k \in D_j} \mathrm{tf}_{kj} \cdot \mathrm{idf}_k^2}} \tag{8}$$

[6] These values can also be considered the ones obtained with the Belief Network Model when simulating the vector space model.

[7] Other probability functions designed for BNRM are shown in [12].

where tf_{ij} is the frequency of the term T_i in document D_j, idf_i is the inverse document frequency of the term T_i in the collection [8] and α is a normalizing constant (to ensure that $\sum_{T_i \in D_j} w_{ij} \leqslant 1 \; \forall D_j \in \mathscr{D}$). This weight has been designed with a similar form to those used in the cosine similarity formula [22].

First, we shall consider the effect of the structure of the term layer, i.e. the set of dependence relationships between terms, on the performance of the system. As mentioned in Section 4.1, we restrict the topology of this subnetwork to a polytree (mainly due to reasons of efficiency). We shall therefore consider two different polytrees that have been obtained using the same learning algorithm: the first where we consider the dependences between all the terms in the collection, $\mathscr{T}$, and the second where we only consider the relationships in a subset of $\mathscr{T}$, $\mathscr{T}^*$, which only contains those terms that can be considered as good discriminators to distinguish between relevant and non-relevant documents. Thus, the terms in $\mathscr{T} \setminus \mathscr{T}^*$ will only be connected to the documents that they belong to. In order to obtain the $\mathscr{T}^*$ set, we consider a frequency-based approach: $T_i \in \mathscr{T}^*$ if the term has a document frequency in the interval $[5, N/10]$, with N being the number of document in the collection. A more detailed study of this problem can be found in [7].

Table 2 shows the average precision for the eleven standard recall points for all the queries of each collection, obtained by SMART [22] and the Inference Network (INM) [24], and the behavior of our model with respect to the particular experimentation.

With respect to INM, we have built our own implementation, and used the configuration parameters proposed by Turtle in [24]:

$$p(t_i|d_j = \text{true}) = 0.4 + 0.6 \cdot \mathrm{tf}_{ij} \cdot \mathrm{idf}_i \quad \text{and} \quad p(t_i|\text{all parents false}) = 0.3 \tag{9}$$

Taking these results into account, we could conclude that the proposed methodology shows a good performance, being comparable with SMART and INM. However, the best results were obtained by considering a particular combination of the parameters: on the one hand, if we fix the structure of the term layer and we study the results of the inclusion (or not) of the 'qf' in the evaluation of the probability function, we can say that the behavior is quite homogeneous, and is clearly dependent on the collections (CACM and CISI support the use of the frequency of query terms whereas CRANFIELD and MEDLARS do not). On the other hand, when fixing the inclusion of the 'qf', we generally obtain better results when considering all the terms in the collection. If we do not include the 'qf', it seems convenient to consider the reduced model.

[8] $\mathrm{idf}_i = \lg \frac{N}{n_i}$, where N is the number of documents in the collection, and n_i is the number of documents that contain the ith term.

Table 2
Experiment with BNRM *with and without* 'qf'

Exp.	SMART	INM	$\mathscr{T}^*$, not 'qf'	$\mathscr{T}^*$ and 'qf'	$\mathscr{T}$, not 'qf'	$\mathscr{T}$ and 'qf'	$P(d\vert Q) - P(d)$
ADI	0.4706	0.4612	0.4632	0.4605	0.4130	0.4613	0.4581
CACM	0.3768	0.3974	0.3692	0.3983	0.3759	0.4046	0.3996
CISI	0.2459	0.2498	0.2104	0.2454	0.2007	0.2301	0.2299
CRAN.	0.4294	0.4367	0.4395	0.4101	0.4314	0.4116	0.4421
MED	0.5446	0.5534	0.6180	0.5764	0.6200	0.5792	0.6407

Our last experiment attempted to discern which method is better for generating the document ranking: sorting the documents according to their probability of relevance, $p(d|Q)$, or by means of the difference of the posterior and prior probabilities, $p(d|Q) - p(d)$. In this case, and in order to reduce the number of experiments, we carried out this last test using all the terms in the collection ($\mathscr{T}$). We also considered the best results obtained in each case, the use of the 'qf' in ADI, CACM and CISI, and CRANFIELD and MEDLARS without using it. The results are presented in the last column of Table 1. Again, it seems that we have a collection dependent behavior: CRANFIELD and MEDLARS perform better when considering the difference of probabilities and the opposite is true for the other collections.

6. Conclusions and future work

In this paper, we have presented an IR model based on Bayesian networks. The topology of the network (qualitative information) that supports the model has been specified, as well as the different probability distributions stored in the nodes (quantitative information). Finally, we have provided it with an inference mechanism to retrieve documents.

IR is a very complex problem due not only to the intrinsic uncertainty related to many aspects of the field, but also because of the size of the problem in terms of the number of documents and terms. We have therefore had to develop several techniques which are able confront the complexity of the problems considered:

- Regarding the learning problem, we propose that the structure be restricted to a type of simplified networks (polytrees) in order to reach acceptable learning and, above all, acceptable inference times. We have also used a specific algorithm which combines useful methodologies from other existing algorithms and incorporates particular features.
- Estimating the qualitative information also presents a very important problem: the huge number of parents that document nodes have in the network. Consequently, this makes any attempt to estimate and later store

the probability distributions impossible. It was for this reason that we developed the probability functions, which allow the probability matrices to be used implicitly.

- The last problem is the inference in our model, because the common methods of propagation in Bayesian networks showed themselves to be totally unable to deal with the large size of the I.R. networks. We therefore designed an inference technique, *Propagation + Evaluation*, which was completely adapted to our topologies. With this mechanism, we obtain some benefits from the specific topology of our network as well as from the probability functions. Consequently, we have been able to put an exact inference in a globally complex network into practice.

The performance of the BNR model clearly depends on the collection as well as the different values of the parameters. For instance, the inclusion of the frequency of the terms of a query in the evaluation of a probability function is good for three collections and not so good for the rest. The same situation arises, when document ranking is carried out, with the use of the difference between probabilities than with only the posterior probability.

The experimentation that we have carried out has attempted to determine the suitability of our new model for document retrieval. In this case, we were able to clearly observe our model's behavior and not be distracted by any other element from our main objective (for instance, the management of greater collections such as those included in TREC). Of course, our next target will be to test our model with real size collections.

With respect to this last matter, the application to this model to very large document collection should suffer some modifications. On the one hand, and considering in a first stage the learning of the polytree with all the terms, as this task must be carry out only once, it is not so important the learning time. Despite this fact, using the idea presented in Section 5, by which only a set of dependences among terms in the collection would be represented in the polytree, the time required to learn this type of graph must be reduced. A more developed selection method than the one presented in this paper is shown in [9], where using a clustering algorithm, terms are divided into two sets: those which are classified as 'good' and those labeled as 'bad', from the point of view of the retrieval. The size of the first class is usually very small, allowing this situation an application of a fast learning, and subsequently also a fast propagation in retrieval time.

On the other hand, and once the polytree has been built, we could apply the techniques presented in [8] to reduce the propagation time. The two approximation methods, modifications of the Pearl's propagation algorithm [18], try to save time by not performing unnecessary inference steps. The reduction of time is considerable with the additional advantage that the loss, in terms of retrieval effectiveness, is almost null. Putting into practice these techniques, the size of the polytree could be very large.

Another of our future lines of research that we are considering is the study of new mechanisms to represent the dependence relationships between terms and/or documents. We are also interested in the development of a method to automatically set up the best values for each parameter of the BNR model. This will be put into practice by analyzing the main characteristics of several collections (idf, document length, term lengths—in the inverted file, and so on) and attempting to obtain common patterns between them.

Acknowledgements

This work has been supported by the Spanish MCYT and FIS, under Projects TIC2000-1351 and PI021147, respectively.

References

[2] N. Christofides, Graph Theory, An Algorithmic Approach, Academic Press, New York, 1975.
[3] F. Crestani, M. Lalmas, C.J. van Rijsbergen, L. Campbell, Is this document relevant?... probably. A survey of probabilistic models in information retrieval, ACM Computing Survey 30 (4) (1991) 528–552.
[4] L.M. de Campos, Independency relationships and learning algorithms for singly connected networks, Journal of Experimental and Theoretical Artificial Intelligence 10 (4) (1998) 511–549.
[5] L.M. de Campos, J.M. Fernández, J.F. Huete, Query expansion in information retrieval systems using a Bayesian network-based thesaurus, in: Proceedings of the 14th Uncertainty in Artificial Intelligence Conference, 1998, pp. 53–60.
[7] L.M. de Campos, J.M. Fernández, J.F. Huete, Reducing term to term relationships in an extended Bayesian network retrieval model, in: Proceedings of the 11th Information Processing and Management of Uncertainty in Knowledge-based Systems, 2002, pp. 543–552.
[8] L.M. de Campos, J.M. Fernández, J.F. Huete, Reducing term to term relationships in an extended Bayesian network retrieval model, in: Proceedings of the First Workshop on Probabilistic Graphical Models, 2002, pp. 35–44.
[9] L.M. de Campos, J.M. Fernández, J.F. Huete, Improving the efficiency of the Bayesian network retrieval model by reducing the relationships among terms, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, in press.
[10] G.F. Cooper, Probabilistic Inference using belief networks is NP-hard, Artificial Intelligence 393–405 (1990) 1990.
[12] J.M. Fernández-Luna, Modelos de recuperación de información basados en redes de creencia, Ph.D. thesis, Universidad de Granada, 2001.
[16] L. Helm, Improbable inspiration, Los Angeles Times, 1996.
[17] M. Indrawan, D. Ghazfan, B. Srinivasan, Using Bayesian networks as retrieval engines, in: Proceedings of the Sixth Text Retrieval Conference, 1996, pp. 437–444.
[18] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan and Kaufmann, San Mateo, CA, 1988.
[19] G. Rebane, J. Pearl, The recovery of causal polytrees from statistical data, in: Uncertainty in Artificial Intelligence, North Holland, Amsterdam, 1989, pp. 175–182.
[20] B.A. Ribeiro-Neto, R.R. Muntz, A belief network model for IR, in: Proceedings of the 19th ACM SIGIR Conference, 1996, pp. 253–260.

[21] S.E. Robertson, The probability ranking principle in IR, Journal of Documentation 33 (4) (1977) 294–304.

[22] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[23] J. Savoy, Bayesian inference networks and spreading activation in hypertext systems, Information Processing & Management 28 (3) (1992) 389–406.

[24] H.R. Turtle, Inference Networks for Document Retrieval Ph.D. Thesis, Computer and Information Science Department, University of Massachusetts, 1990.

[25] H.R. Turtle, W.B. Croft, Efficient probabilistic inference for text retrieval, in: Proceedings of the RIA0'91 Conference, 1991, pp. 644–661.

[26] H.R. Turtle, W.B. Croft, Uncertainty in information retrieval systems, in: Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer Academic Publishers, Dordrecht, 1997, pp. 189–224.

[28] C.J. van Rijsbergen, Information Retrieval, second ed., Butter Worths, London, 1979.