A Bayesian Approach to User Profiling in Information Retrieval

S.K.M. Wong
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
wong@cs.uregina.ca

C.J. Butz
School of Information Technology & Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
butz@site.uottawa.ca

Abstract

Numerous probability models have been suggested for *information retrieval* (IR) over the years. These models have been applied to try to manage the inherent uncertainty in IR, for instance, document and query representation, relevance feedback, and evaluating the effectiveness of IR system. On the other hand, *Bayesian networks* have become an established probabilistic framework for uncertainty management in artificial intelligence.

In this paper, we suggest the use of Bayesian networks for user profiling in IR. Our approach can take full advantage of both the effective learning algorithms and efficient query processing techniques already developed for probabilistic networks. Moreover, Bayesian networks capture a more general class of probability distributions than the previously proposed probabilistic models. Finally, this paper provides a theoretical foundation for the cross-fertilization of techniques between IR and Bayesian networks.

1 Introduction

In the environment of *information retrieval* (IR) [8, 24], there exists a collection of documents and a group of users. The primary objective of a retrieval system is to identify those documents that are useful to a particular user. A document is *relevant* to a user whenever it is thought that the document is useful, otherwise the document is considered as being *nonrelevant.* There are many complex factors which govern the relevance relationship between a document and a user query. It is practically impossible to design a system that will make strict predictions about such relevance relationships. However, the problem of retrieval can be formulated as a problem of evidence and prediction based on the probability theory [19].

Traditional probabilistic models construct a discriminant (decision) function, representing the information request, through an inductive learning process (relevance feedback). Although the probabilistic model is theoretically sound, and its retrieval effectiveness extensively studied [4, 5, 6, 18, 29], one is often forced to make some rather restrictive assumptions on the joint probability distribution. For example, due to the problem of large dimensionality, we may assume that the index terms are probabilistically independent. The independence model is simple but its validity is questionable. Other higher-order approximations [18, 24, 29, 30] were suggested such as the tree dependence model [7]. However, most of these approaches can only capture a restricted subclass of probability distributions.

Turtle and Croft [22, 23] introduced an inference model for reasoning with uncertainty to IR. In [10], this model was extended to include relevance feedback. More recently, Fung and Del Favero [9] suggested a method for applying *Bayesian networks* [17] to IR. All of these approaches are *online* methods which construct a network for each individual query. Such an approach may not always be practical. Thus, we suggest an alternative approach of applying Bayesian networks to IR.

In this paper, we propose a method for constructing a user profile using either a Bayesian or Markov network. The input to our approach is a sample of documents that the user has marked as either relevant or nonrelevant. We can then learn a probabilistic network which encodes the user's preferences. Such a network provides a formal foundation for probabilistic inference. Documents can then be ranked according to the conditional probability defined by the network. Our approach has several advantages. We would like to emphasize that the class of probability distributions represented in the Chow and Liu method [7] is a proper subset of the distributions represented in our approach. Our method can take full advantage of the established techniques already employed for uncertainty management in artificial intelligence. For instance, many algorithms exist for learning a Bayesian network [3, 12, 13, 15, 21] or a Markov network [27, 28]. Moreover, efficient inference techniques exist for query processing in Bayesian networks [17] and Markov networks [14, 16, 20]. In this discussion, we promote the harmonization of the IR and Bayesian network communities by *directly* adopting the proven learning and query processing techniques already implemented in probabilistic reasoning systems.

We would like to make it clear that the work here is quite different from [9, 10, 22, 23]. As already mentioned, those methods take an *online* approach. That is, a network is built at the time the query is issued by the user. On the contrary, we learn our network *offline*. Our approach should give a faster response time since the network used online is already fixed. The probabilistic network can always be *refined* based on new samples that the user has marked as either relevant or nonrelevant. This means that the quality of the probabilistic network used in practice will improve as the sample size increases.

This paper is organized as follows. In Section 2, we review the two types of probabilistic networks, namely, Bayesian networks and Markov networks. In Section 3, we review Chow and Liu's [7] pioneering work on learning probabilistic networks. We show how this work can be generalized in Section 4. In Section 5, we discuss how these general learning algorithms can be applied to build a user profile. The conclusion is given in Section 6.

2 Probabilistic Models

In this section, we introduce two frameworks for the representation of probabilistic knowledge, namely, Bayesian networks and Markov networks [2, 11, 17].

2.1 Bayesian Networks

Let $R = \{A_1, A_2, \ldots, A_k\}$ denote a finite set of discrete variables. Each variable A_i is associated with a finite domain D_i. Let D be the Cartesian product of the domains D_i, $1 \leq i \leq k$. A *joint probability distribution* (jpd) [11, 17] on D is a function p on D, assigning to each tuple $t \in D$ a real number $0 \leq p(t) \leq 1$ such that $\sum_{t \in D} p(t) = 1$. In general, a *potential* [11] is a function q on D such that $q(t)$ is a nonnegative real number and $\sum_{t \in D} q(t)$ is positive, i.e., at least one $q(t) > 0$. Each potential q can be transformed to a joint probability distribution p through *normalization*, that is, by setting $p(t) = q(t)/\sum_{v \in D} q(v)$. We say the distribution p is on R, and sometimes write p as $p(R)$, if the domain D is understood.

We say Y and Z are *conditionally independent* given X under p, denoted $I_p(Y, X, Z)$, if

$$p(y \mid x, z) \quad = \quad p(y \mid x), \tag{1}$$

whenever $p(x, z) > 0$. This conditional independency $I_p(Y, X, Z)$ can be equivalently written as

$$p(y, x, z) \quad = \quad \frac{p(y, x) \cdot p(x, z)}{p(x)}. \tag{2}$$

We write $I_p(Y, X, Z)$ as $I(Y, X, Z)$ if the joint probability distribution p is understood. In the special case where $Y \cup X \cup Z = R$, we call the probabilistic conditional independence $I(Y, X, Z)$ *nonembedded*; otherwise $I(Y, X, Z)$ is called *embedded*.

By the chain rule, a joint probability distribution p over $R = \{A_1, A_2, \ldots, A_m\}$ can always be written as:

$$\begin{aligned} p(R) \quad = \quad & p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1, A_2) \cdot \ldots \\ & \cdot p(A_m|A_1, A_2, \ldots, A_{m-1}). \end{aligned} \tag{3}$$

The above equation is an identity. However, one can use conditional independencies that are assumed to hold in the problem domain to obtain a simpler representation of a joint distribution. For example, consider a joint distribution $p(A, B, C, D, E, F)$. By equation (3), the joint distribution $p(R)$ can be written as

$$\begin{aligned} p(R) \quad = \quad & p(A) \cdot p(B|A) \cdot p(C|A, B) \\ & \cdot p(D|A, B, C) \cdot p(E|A, B, C, D) \\ & \cdot p(F|A, B, C, D, E). \end{aligned} \tag{4}$$

Consider the following set $\mathbf{C}$ of conditional independencies:

$$\begin{aligned} p(C|A, B) \quad &= \quad p(C|A), \\ p(D|A, B, C) \quad &= \quad p(D|B, C), \\ p(E|A, B, C, D) \quad &= \quad p(E|B, C), \\ p(F|A, B, C, D, E) \quad &= \quad p(F|E), \end{aligned}$$

namely,

$$\begin{aligned} \mathbf{C} \quad = \quad & \{I(C, A, B),\ I(D, BC, A), \\ & I(E, BC, AD),\ I(F, E, ABCD)\}, \end{aligned} \tag{5}$$

where $A_i \ldots A_j$ denotes $\{A_i, \ldots, A_j\}$ Utilizing these conditional independencies, the joint distribution $p(R)$ written using the chain rule in equation (4) can be expressed in a simpler form, namely:

$$\begin{aligned} p(R) \quad = \quad & p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B, C) \\ & \cdot p(E|B, C) \cdot p(F|E). \end{aligned} \tag{6}$$

We can represent the dependency structure of this joint distribution by the *directed acyclic graph* (DAG) shown in Figure 1. This DAG, together with the conditional probability tables $p(A)$, $p(B|A)$, $p(C|A)$, $p(D|B,C)$, $p(E|B,C)$, and $p(F|E)$, define a Bayesian network. Such a network provides an economical representation of a joint probability distribution.

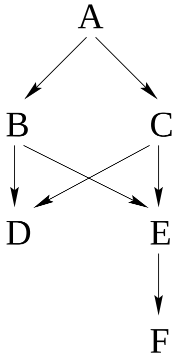

Figure 1: The directed acyclic graph reflecting the conditional independencies defined in equation (6).

2.2 Markov Networks

Even though Bayesian networks provide an economical representation of a joint probability distribution, it may still be difficult to compute marginal distributions. Thus, several efficient local computation algorithms [14, 16, 20] were developed for computing marginal distributions in *Markov networks* [11]. It should be noted that a Markov network defined by Hajek, Havranek and Jirousek [11] is called a *decomposable* Markov network by Pearl [17]. That is, the definition of Markov network in [17] is different from the one used here.

To facilitate probabilistic inference, it is useful to transform a Bayesian network into a Markov network. The DAG representing the dependency structure of a Bayesian network can be converted by the moralization and triangulation procedures [11, 17] into an *acyclic hypergraph.* (An acyclic hypergraph in fact represents a chordal undirected graph. Each maximal clique in the graph corresponds to a hyperedge in the acyclic hypergraph.) For example, by applying these procedures to the DAG in Figure 1, we obtain the acyclic hypergraph depicted in Figure 2. Such an acyclic hypergraph represents the dependency structure of a Markov network. The joint probability distribution defined by equation (6) can be rewritten in terms of marginal distributions as:

$$\begin{aligned} &p(R) \\ =\ & \frac{p(A,B,C)\cdot p(B,C,D)\cdot p(B,C,E)\cdot p(E,F)}{p(B,C)\cdot p(B,C)\cdot p(E)}. \end{aligned} \tag{7}$$

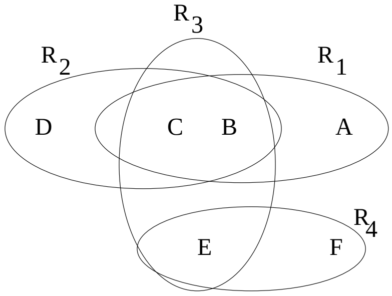

Figure 2: A graphical representation of the acyclic hypergraph $\mathcal{H} = \{\ R_1 = \{A,B,C\},\ R_2 = \{B,C,D\},\ R_3 = \{B,C,E\},\ R_4 = \{E,F\}\}$.

A Bayesian network is more expressive than a Markov network. Consider the marginal distribution $p(A,B,C)$ obtained from the Bayesian network in equation (6):

$$\begin{aligned} & p(A,B,C) \\ =\ & \sum_{D,E,F} p(A,B,C,D,E,F) \\ =\ & \sum_{D,E,F} p(A)\cdot p(B|A)\cdot p(C|A)\cdot p(D|B,C) \\ & \qquad \cdot p(E|B,C)\cdot p(F|E) \\ =\ & p(A)\cdot p(B|A)\cdot p(C|A) \\ =\ & \frac{p(A,B)\cdot p(A,C)}{p(A)}. \end{aligned}$$

Thus, the *embedded* probabilistic conditional independence $p(C|A,B) = p(C|A)$ is reflected by the Bayesian network in equation (6). However, this same independency is *not* reflected by the Markov network in equation (7):

$$\begin{aligned} & p(A,B,C) \\ =\ & \sum_{D,E,F} p(A,B,C,D,E,F) \\ =\ & \frac{p(A,B,C)\cdot p(B,C,D)\cdot p(B,C,E)\cdot p(E,F)}{p(B,C)\cdot p(B,C)\cdot p(E)} \\ =\ & \frac{p(A,B,C)\cdot p(B,C)\cdot p(B,C)}{p(B,C)\cdot p(B,C)} \\ =\ & p(A,B,C). \end{aligned}$$

In the above discussion, we examined two types of probabilistic networks, namely, Bayesian and Markov. A Bayesian network is defined by a DAG and corresponding conditional probability distributions. On the other hand, a Markov network is defined by an acyclic hypergraph and corresponding marginal distributions.

3 Learning a Dependence Tree

Chow and Liu [7] developed an elegant method to approximate an n-dimensional discrete probability distribution by a product of second-order distributions. The conditional independencies learned by their algorithm are represented by a singly-connected DAG, called a *first-order dependence tree* in [7].

The singly-connected DAG in Figure 3 (i) was given in [7] as an example of a (first-order) dependence tree. This dependence tree indicates that the joint distribution can be written as:

$$\begin{aligned} & p(ABCDEF) \quad (8) \\ = \; & p(A) \cdot p(B|A) \cdot p(C|B) \cdot p(D|B) \cdot p(E|B) \cdot p(F|E). \end{aligned}$$

(We have reversed the direction of the arrows to make it consistent with Bayesian networks.) The factorization of the joint distribution $p(ABCDEF)$ in Equation (8) can be equivalently written in terms of marginal distributions:

$$\begin{aligned} & p(A, B, C, D, E, F) \quad (9) \\ = \; & \frac{p(A, B) \cdot p(B, C) \cdot p(B, D) \cdot p(B, E) \cdot p(E, F)}{p(B) \cdot p(B) \cdot p(B) \cdot p(E)} \end{aligned}$$

The undirected graph in Figure 3 (ii) expresses exactly the same conditional independencies as the DAG in Figure 3 (i). The undirected graph in Figure 3 (ii) can be equivalently represented as an acyclic hypergraph $\mathcal{H}$, as illustrated in Figure 3 (iii). (Recall that the maximal cliques in the undirected graph are precisely the hyperedges of $\mathcal{H}$.)

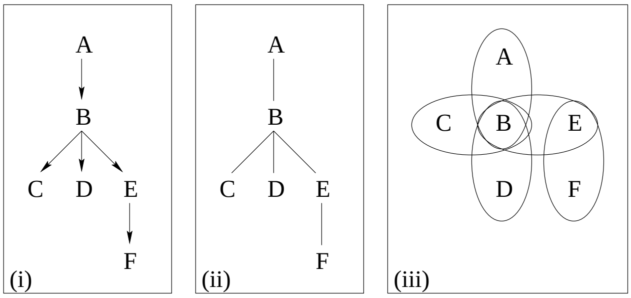

Figure 3: (i) A singly-connected DAG (a dependence tree) $\mathcal{D}$. (ii) The corresponding undirected graph $\mathcal{U}$ representing precisely the same independency information as $\mathcal{D}$. (iii) The undirected graph $\mathcal{U}$ in (ii) can be equivalently represented as an acyclic hypergraph $\mathcal{H}$.

The important point is that their method is guaranteed to find an *optimal* approximation of the joint distribution under the given scoring metric (a measure of closeness) and the restriction to using second-order distributions. Hence, one cannot improve their method under these conditions. However, in the next section, we discuss two ways in which their method can be generalized for the purpose of learning a user profile.

4 Learning Probabilistic Models

Learning Markov and Bayesian networks are both generalizations of the Chow and Liu method.

As already mentioned, the conditional independencies learned by the Chow and Liu method are represented by a singly-connected DAG. Every singly-connected DAG is equivalent to a (decomposable) *Markov network*. More specifically, every singly-connected DAG is equivalent to a Markov network in which every marginal distribution is defined on precisely two variables. In other words, the graphical structure learned by Chow and Liu is a subclass within the more general class of Markov networks. This means that, by relaxing the condition that every marginal distribution must involve precisely two variables, one can learn *any* Markov network. In fact, we have already developed a method which can learn a Markov network from sample data [27, 28]. Experimental results and complexity analysis can be found in [27, 28].

A second way to generalize the Chow and Liu method is to also learn *embedded* CIs. Recall that a Markov network only encodes *full* CIs, namely, those CIs which involve every variable in the network. On the other hand, an embedded CI is a CI which is allowed to involve a proper subset of variables in the network (see Section 2.1). In other words, full CIs are a special case of embedded CIs; an embedded CI becomes a full CI exactly when the CI involves every variable in the network. Many learning algorithms exist for learning Bayesian networks from data [3, 12, 13, 15, 21].

5 Building the User Profile

We suggest that a user profile can be represented as a probabilistic network. Such a representation offers many advantages. A probabilistic network provides a formal foundation for probabilistic inference. More importantly, queries involving *any* subset of terms (attributes) may be posed to the network. Finally, we can *directly* employ the existing techniques already implemented in probabilistic reasoning systems for IR purposes, namely, the well-studied learning algorithms and the proven query optimization techniques.

The input to our approach is a sample of documents, represented by a fixed set of terms, that the user has marked as either relevant or nonrelevant. (Note that the learning algorithms do not depend on binary valued attributes.) Thus, one attribute labelled *Relevance* can be appended to the usual vector representation. This sample data can be given as input to our Markov network learning algorithm [27, 28], or any one of the Bayesian network learning algorithms [3, 12, 13, 15, 21]. The learning algorithms will treat the additional column *Relevance* as simply another column. Thereby, the output will be either a Markov or Bayesian network with the exception that one attribute in the network represents the preference of the user. Once the probabilistic network is constructed, the documents can be ranked according to the computed conditional probabilities. More importantly, perhaps, queries can be posed to the network involving *any* subset of terms.

It is important to note that the probabilistic network can be *refined.* The user can mark subsequently viewed documents as either relevant or nonrelevant. These new samples can be used to modify the network in an offline mode. In other words, these new samples can be added to the original sample. The quality of the learned network will increase as the sample size gradually increases. We illustrate the proposed method for user profiling with the aid of the following example.

Consider a user who receives numerous *electronic mail* (email) messages each day. This particular user is too busy to read all of the new email messages received each day. Thereby, she would prefer to read the most relevant email messages of the unread messages. For simplicity, let us assume in this example that every email message is represented by the same fixed set $\{A_1, A_2, \ldots, A_n\}$ of terms. Suppose further that there is available an auto-indexing program which will assign values $A_1 = a_1$, $A_2 = a_2$, …, $A_n = a_n$ to each newly arrived email message. Given a sample of email messages that the user has marked as either relevant or nonrelevant, we can apply the learning algorithms discussed in Section 4 to learn a Bayesian or Markov network. Whenever a new email message arrives, we rank it according to the following conditional probability defined by the probabilistic network:

$$p(Rel = relevant \mid A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n).$$

where *Rel* stands for *Relevance.* Based on the original probabilistic network, let $e_1, e_2, e_3, e_4, e_5, e_6$ be (the vector representations of) six new email messages with conditional probabilities:

$$\begin{aligned}
p(Relevance = relevant \mid e_1) &= 0.5,\\
p(Relevance = relevant \mid e_2) &= 0.1,\\
p(Relevance = relevant \mid e_3) &= 0.7,\\
p(Relevance = relevant \mid e_4) &= 0.0,\\
p(Relevance = relevant \mid e_5) &= 0.9,\\
p(Relevance = relevant \mid e_6) &= 0.3.
\end{aligned}$$

Thereby, these new email messages would be ranked as $e_5, e_3, e_1, e_6, e_2, e_4$. Following this ranking, let us assume that the user has time to read e_5, e_3, e_1, which she ranks as *relevant*, *nonrelevant*, and *relevant*, respectively. Using these three new samples, the original probabilistic network can be *refined* in an offline mode. Suppose that the ranking, specified by the refined network, of the previously unread messages e_6, e_2, e_4 and the newly arrived messages e_7, e_8 is:

$$\begin{aligned}
p(Relevance = relevant \mid e_8) &= 0.9,\\
p(Relevance = relevant \mid e_6) &= 0.7,\\
p(Relevance = relevant \mid e_2) &= 0.2,\\
p(Relevance = relevant \mid e_7) &= 0.1,\\
p(Relevance = relevant \mid e_4) &= 0.0.
\end{aligned}$$

Notice that the refined network defines a different conditional probability for the messages e_6 and e_2.

6 Conclusion

In this paper, we have suggested that a user profile can be represented as either a Bayesian or Markov network. Such a network is learned from a sample of documents that are judged by the user to be relevant or nonrelevant. As probabilistic networks are well-established as a rigorous foundation for uncertainty management [11, 17], we can process queries posed to the network involving any subset of index terms taking on any values from their respective domains. Moreover, the probabilistic network can be *refined* as the user views new documents. In other words, as the sample size increases, so does the quality of the learned network.

Our approach has several advantages. The probability distributions represented in the Chow and Liu method [7] form a subclass of distributions represented by Bayesian and Markov networks. Our method can take full advantage of the established techniques already employed for uncertainty management in artificial intelligence. For instance, many algorithms exist for learning a Bayesian network [3, 12, 13, 15, 21] or a Markov network [27, 28]. Moreover, efficient inference techniques exist for query processing in Bayesian networks [17] and Markov networks [14, 16, 20]. Since we *directly* adopt techniques already implemented in probabilistic reasoning systems, the discussion here can then be seen as a theoretical foundation for harmonizing the IR and Bayesian network communities.

Finally, we would like to emphasize that the work here is quite different from other proposed methods of applying inference networks for IR purposes [9, 10, 22, 23]. Those methods build a network *online* for each query issued by the user. On the contrary, we learn our network *offline*. Our approach should give a faster response time since the network used in practice is already fixed.

References

[1] C. Beeri, R. Fagin, D. Maier and M. Yannakakis, On the desirability of acyclic database schemes. *JACM*, **30**, 3, 479-513, 1983.

[2] C.J. Butz, The relational database theory of Bayesian networks. Ph.D. Thesis, Department of Computer Science, University of Regina, 2000.

[3] G.F. Cooper and E.H. Herskovits, A Bayesian method for the induction of probabilistic networks from data, *Machine Learning*, **9**, 309-347, 1992.

[4] F. Crestani and C.J. van Rijsbergen, Probability kinematics in information retrieval, *Proceedings of the 18th Annual International ACM-SIGIR conference*, 291-299, 1995.

[5] F. Crestani and C.J. van Rijsbergen, A study of probability kinematics in information retrieval, *ACM Transactions on Information Systems*, **16**, 3, 225-255, 1998.

[6] W.B. Croft and D.J. Harper, Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, **35**, 106-119, 1977.

[7] C.K. Chow and C.N. Liu, Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **IT-14**, 3, 462-467, 1968.

[8] Frakes, W., Baeza-Yates, R. (Eds.), *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, USA, 419-442, 1992.

[9] R. Fung and B. Del Favero, Applying Bayesian networks to information retrieval. *Communication of ACM*, **38**, 3, 42-48,57, 1995.

[10] D. Haines and W.B. Croft, Relevance feedback and inference networks. *Proceedings of the 16th Annual International ACM-SIGIR conference*, 2-11, 1993.

[11] P. Hajek, T. Havranek and R. Jirousek, *Uncertain Information Processing in Expert Systems*. CRC Press, 1992.

[12] D. Heckerman, D. Geiger and D.M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, **20**, 197-243, 1995.

[13] E.H. Herskovits and G.F. Cooper, Kutato: an entropy-driven system for construction of probabilistic expert systems from database. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 54-62, 1990.

[14] F.V. Jensen, S.L. Lauritzen and K.G. Olesen, Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, **4**, 269-282, 1990.

[15] W. Lam and F. Bacchus, Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, **10**, 3, 269-293, 1994.

[16] S.L. Lauritzen and D.J. Spiegelhalter, Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, **B**, 50, 157-244, 1988.

[17] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, California, 1988.

[18] S.E. Robertson and K. Sparck Jones, Relevance weighting of search terms. *Journal of the American Society for Information Science*, **27**, 129-146, 1976.

[19] S.E. Robertson, The probability ranking principle in IR. *Journal of Documentation*, **33**, 294-304, 1977.

[20] G. Shafer, An axiomatic study of computation in hypertrees. *Technical report*. University of Kansas, School of Business Working Papers, 232, 1991.

[21] P. Spirtes and C. Glymour, An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, **9**, 1, 62-73, 1991.

[22] H.R. Turtle, Inference networks for document retrieval. Ph.D. Thesis, University of Massachusetts, 1990.

[23] H.R. Turtle and W.B. Croft, Inference networks for document retrieval. *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, 1-24, 1990.

[24] C.J. van Rijsbergen, *Information Retrieval*. Butterworths, London, UK, 1979.

[25] S.K.M. Wong and Y.Y. Yao, A probability distribution model for information retrieval. *Information Processing & Management*, **25**, 1, 39-53, 1989.

[26] S.K.M. Wong and Y.Y. Yao, A generalized binary probabilistic independence model. *Journal of the American Society for Information Science*, **41**, 5, 324-329, 1990.

[27] S.K.M. Wong and Y. Xiang, Construction of a Markov network from data for probabilistic inference. *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, 562-569, 1994.

[28] Y. Xiang, S.K.M. Wong and N. Cercone, A "microscopic" study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, **26**, 1, 65-92, 1997.

[29] C.T. Yu and G. Salton, Precision weighting - an effective automatic indexing method. *Journal of the Association for Computing Machinery*, **23**, 76-88, 1976.

[30] C.T. Yu, W.S. Luk and T.Y. Cheung, A statistical model for relevance feedback in information retrieval. *Journal of the Association for Computing Machinery*, **23**, 273-286, 1976.