第32卷 第15期 计 算 机 工 程 2006年8月
Vol.32 № 15 Computer Engineering August 2006
·软件技术与数据库· 文章编号：1000—3428(2006)15—0067—03 文献标识码：A 中图分类号：TP393

# 基于贝叶斯网络的XML文档查询模型

徐建民[1,2]，柴变芳[1]，姚冬磊[3]，赵 爽[1]

(1. 河北大学数学与计算机学院，保定 071002；2. 天津大学系统与工程研究所，天津 300072；3. 河北省财税信息中心，石家庄 050051)

**摘 要：**目前XML查询语言及查询界面对Web用户过于复杂，该文描述了一种XML文档索引机制，在此基础上建立了一个通用的贝叶斯网络查询模型。用户只需在交互界面输入自然语言描述的查询，系统就能对其实现基于语义的构造，由它生成多个结构化查询；对这些查询建立贝叶斯网络，计算查询在给定文档下的概率，选择概率最大的前3个查询提交给系统执行。

**关键词：**贝叶斯网络；信息检索；结构化查询；XML

# Query Model of XML Document Based on Bayesian Network

XU Jianmin[1,2], CHAI Bianfang[1], YAO Donglei[3], ZHAO Shuang[1]

(1 . College of Mathematics and Computer, Hebei University, Baoding 071002; 2. Institute of Systems Engineering, Tianjin University, Tianjin 300072; 3. Finance Information Center of Hebei Province, Shijiazhuang 050051)

**【Abstract】**Recently, the query languages and interfaces for XML are too intricate for Web users. This paper presents an index mechanism for XML documents, builds a retrieval model based on Bayesian network. The model constructs the user's query based on semantic in natural language, generates several structured queries, and makes Bayesian network for queries, and computes the probabilities on the document collection. Then the system selects the top there queries to execute.

**【Key words】** Bayesian network; Information retrieval; Structured query; XML

互联网上资源各种各样，为实现简单、方便、有效的Web查询，越来越多的信息采用XML数据模型统一描述Web上的各种资源，XML将成为Internet上数据描述和交换的标准，并且将会代替HTML而成为Web上驻留数据的主要格式。XML的出现为信息的处理提供了内容与结构两方面强有力的支持，对XML文档信息检索的研究也越来越受到重视。

国外主要从2方面对XML检索进行研究：(1)设计XML查询语言；(2)对传统信息技术进行改进。前者一般基于路径和树模式，需要最终用户熟悉查询语言的语法，并要求用户全面了解文档结构，这与Web查询界面简单一致的原则相违背。后者把文档看成是一堆关键词的集合，不考虑或很少考虑文档的结构信息以及语义信息，用于XML文档的检索不能很好地利用XML文档的结构信息及其固有的语义信息[1]。

针对上述问题，本文设计了一个简单准确的基于贝叶斯网络的查询构造模型：用户只需要输入简单的自然语言，系统根据XML文档的内容和结构生成多个结构化查询，利用它们建立贝叶斯网络，计算各个查询在给定文档集合下的概率，选择概率最大的3个查询执行。

## 1 贝叶斯网络

贝叶斯网络在过去15年应用于信息检索来解决不确定知识，为支持排序的各种证据建模提供了一个有效灵活的框架。它可以表示术语间的条件概率和概念语义，并依此预测用户查询和文档间的相似度，是解决信息检索领域问题的有效手段。贝叶斯网络[2,3]，又叫信念网络，是由节点、有向弧和条件概率分布组成的有向无环图。节点代表领域中的变量，有向弧表示变量之间的因果关系，变量之间的关系强弱由节点与其父节点之间的条件概率来表示。通过贝叶斯网络可以准确地反映实际应用中变量之间的依赖关系。

在信息检索领域，主要是利用贝叶斯网络表示术语间的关系以及对查询与文档间的相似度进行预测，从而实现基于语义概念的查询[4]。一般把查询、文档和术语看作事件，这些事件相互依赖。如某个术语出现次数将影响文档的出现。

假设文档和查询都包含关键词，用贝叶斯网络建立文档检索模型如图1所示。

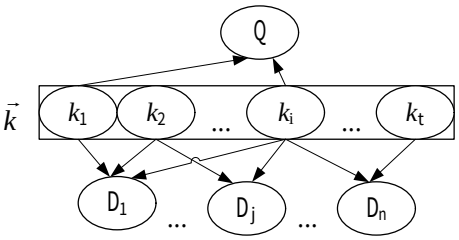


**图1 信息检索中的贝叶斯网络**

图1中 $k_i$表示系统中的术语，每个术语和一个二进制随机变量相关，也用$k_i$表示。向量 $\vec{k}=(k_1,...,k_t)$，其中$k_i$为随机变量，$k_i$=1 表示相应术语$k_i$出现在 $\vec{k}$ 对应的概念中，$\vec{k}$ 有 $2^t$种取值。用 $g_j(\vec{k})$ 表示 $\vec{k}$ 中变量$k_i$的值。

$Q$表示用户查询，$Q=\{k_1',k_2',...,k_t'\}$，其中 $k_j'$ 表示和术语$k_i$相关的二进制随机变量，$g_j(Q)$ =1 表示术语$k_i$出现在查询$Q$中，否则不出现在$Q$中。

$D_j$表示文档，$D_j=\{k_1,k_2,...,k_t\}$，其中$k_i$表示和术语$k_i$相关

**基金项目：**河北省科学技术研究与发展基金资助项目(04213534)
**作者简介：**徐建民(1966—)，男，博士生、教授，主研方向：信息处理；柴变芳，硕士生；姚冬磊，学士；赵 爽，硕士生
**收稿日期：**2005-08-22 **E-mail：**wgwei@21cn.com

的二进制随机变量，$g_j(D_j)=1$表示术语$k_i$出现在文档$D_j$中，否则不出现在$D_j$中。

通过计算查询和文档的相似程度$P(D_j|Q)$，概率值最大的文档将最满足用户需要。

## 2 XML 文档处理

如图 2 所示，一篇 XML 文档是按层次化的树型结构组织的，由若干章节组成，每一章节由若干段组成。它可被转换成一棵对象节点树，在这棵树中，根节点代表文档，其他的章、段落都是根节点的后代节点(称为元素)，叶节点表示元素内容。

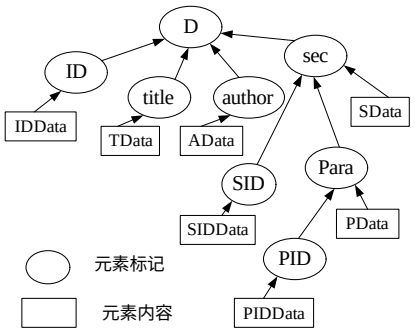


**图 2 XML 文档树**

优化的索引技术是建立查询的基础，文中的查询模型是基于现描述的索引机制[5]。在XML文档中，元素文本的索引为(词汇，出现次数，位置)，其中位置结构为(文档ID，章ID，段ID)。查询之前对文档进行如下处理：从文档树中的所有节点中抽取索引词条，可将文档树处理为词汇树；词汇树中某个层次上的某个节点中的某个词汇的权重超过一定的阈值，应该将其从它所在的节点中删除，提升到高一层上，从文档底层到顶层处理所有节点，最后可以把整个文档整合为一个庞大的汇总树，根节点为文档集合，不包含概念；对汇总树各结构单元建立倒排文档。

## 3 XML 文档查询过程

### 3.1 概念描述

这里只考虑 XML 文档的抽象描述，不考虑特殊的存储，具有普遍性。

**定义 1** 文档$D=\{D_1, D_2,..., D_n\}, n\geq 1$。每个文档$D_i=\{SN,TN\}$，$D_i$由结构化单元集合和文本信息集合组成。$SN=\{s_d^1,...,s_d^{|d|}\}$，其中$s_d^i$表示文档的第 i 个结构单元，结构单元按照从上到下、从左到右排序。对于每个文档，它的文本信息为集合$TN=\{t_d^1,...,t_d^{|d|}\}$，$|d|$是文档结构单元的个数。每个结构单元$s_d^i$对应的文本信息$t_d^i$是由一些概念组成的$t_{ij}$，$t_d^i=\{t_{i1},...,t_{in_i}\}$，$n_i$表示$t_d^i$中的概念个数。

**定义 2** 按照索引机制将所有文档建立成一个词汇树，各结构单元包含的术语$\tau=\{T_1, T_2, ..., T_n\}$，其中$T_i$表示第 i 个结构单元对应的术语集合。

**定义 3** 非结构化查询$U=\{c_1, c_2, ..., c_k\}$，其中$c_i$表示概念，由词或短语组成。

**定义 4** 结构化查询 Q 由顺序对组成的集合：

$$Q=\{<s_d^1,c_{11}>,<s_d^1,c_{12}>,...,<s_d^1,c_{1n1}>,...,<s_d^{|d|},c_{|d|1}>,...,<s_d^{|d|},c_{|d|n_{|d|}}>\}$$

其中$s_d^i$是第 i 个结构单元，它可以有多个指向，$c_{ij}$是出现在$s_d^i$中的概念。

### 3.2 查询过程

在 Web 用户对需要的信息不太了解的情况下，查询的交互界面根据实际情况给出用户提示。用户在交互界面文本框中输入查询内容，然后对查询内容进行以下处理：

(1)对用户输入进行自然语言处理：借助知识库(包括语法、句法知识，语义、语用知识，常识，语料库，词典数据库，禁用词表和反向词频统计表)里的词法、句法知识、分词词典，利用设定的程序对用户输入的查询语句进行自动分词，获得能正确表达查询意义的概念性词或词组，以此作为查询的基本概念去检索文档集合。

(2)查询扩展：基于同义词典和相关词典，对提取的概念进行语义扩展。为避免检索结果不准确，引入反馈机制，用户得到初次查询结果，反馈调整意见，系统自动调整语义扩展算法，通过多次学习系统会得到更好的结果。

(3)构造非结构化查询：参照从扩展后的查询分析出非结构化查询$U=\{c_1, c_2, ..., c_k\}$。

(4)构造结构化查询：结合非结构化查询U和术语集合$T_i$，得出多个结构化查询。

(5)建模：将结构化查询、文档汇总树和术语集合$\tau$建立贝叶斯网络模型，计算每个结构化查询在给定数据库下的概率，选择概率最大的结构化查询提交。

## 4 基于贝叶斯网络的 XML 数据库查询模型

### 4.1 贝叶斯网络结构

按照第 3 节把 XML 文档集合分析为汇总树，为了描述方便，假设 XML 文档汇总树包含 3 个结构单元。如图 3，每个节点表示要解决问题域中一个变量，变量值为二进制，当计算概率时变量相关则值为 1，否则为 0。

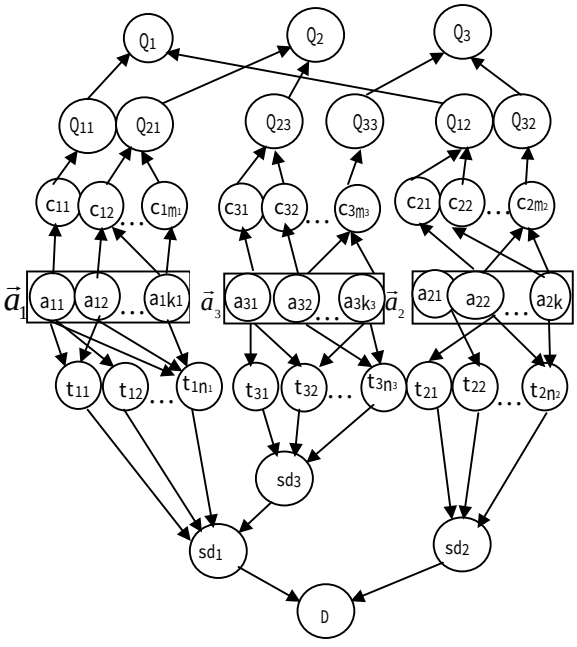


**图 3 基于贝叶斯网络的 XML 数据库查询模型**

D为文档集合根节点，结构单元为$sd_1$、$sd_2$、$sd_3$；$t_{ij}$表示第i个结构单元包含的概念；$a_{ij}$表示$T_i$中的一个术语，向量$\vec{a}_i$表示术语集$T_i$中的一种取值情况，它有$2^{|T_i|}$种取值；$c_{ij}$表示和$Q_{qi}$相关的概念；$Q_i$表示要排列的结构化查询，$Q_{ik}$表示$Q_i$和第k个结构单元对应部分查询。

### 4.2 推理

$P(Q_i|D)$表示在给定文档集合下 D 得到查询$Q_i$的概率，令向量$\vec{k}=\vec{a}_1,\vec{a}_2,\vec{a}_3$：

$$P(Q_i \mid D) = \eta \sum_{\vec{k}} P(Q_i \mid \vec{k}) P(D \mid \vec{k}) P(\vec{k}) \tag{1}$$

其中 $\eta$ 为常量，实例化式(1)，则

$$P(Q_i \mid D) = \eta \sum_{\vec{k}} P(Q_i \mid Q_{i1}, Q_{i2}, Q_{i3}) P(Q_{i1}, Q_{i2}, Q_{i3} \mid \vec{k}) P(D \mid sd_1, sd_2, sd_3) P(sd_1, sd_2, sd_3 \mid \vec{k}) P(\vec{k}) \tag{2}$$

$$P(Q_i \mid Q_{i1}, Q_{i2}, Q_{i3}) = \frac{C^{k_i}}{C^{k_U}} \quad \text{iff} \quad Q_{ik} \text{同时为 } 1 \tag{3}$$

$Q_{ik}$为$Q_i$包含的部分查询( $k \in [1,2,3]$)，$P(Q_i \mid Q_{i1}, Q_{i2}, Q_{i3})$ 等于$Q_i$在它包含的部分查询下的概率。当$Q_i$包含的部分查询只有 1 个或 2 个时，此部分概率通过$Q_i$在它的部分查询下的概率来计算；$k_i$表示结构化查询$Q_i$包含的术语数；$k_U$表示用户给定非结构化查询包含的术语数；C是经验常数，$C \geq 1$。

$$P(D \mid sd_1, sd_2, sd_3) = 1 \quad \text{iff } sd_k \text{ 同时为 } 1 \tag{4}$$

$sd_k$ 表示和查询$Q_{ik}$对应的术语相关层元素，( $k \in [1,2,3]$)，如果网络中$Q_{ik}$相关，而$sd_k$不相关，则 $P(D \mid sd_1, sd_2, sd_3) = 0$。

由于向量 $\vec{a}_1, \vec{a}_2, \vec{a}_3$ 相互独立，则

$$P(Q_{i1}, Q_{i2}, Q_{i3} \mid \vec{k}) = \prod_{j=1}^{3} \mathrm{P}(\mathrm{Q_{ij}} \mid \vec{a}_j)$$

$$P(sd_1, sd_2, sd_3 \mid \vec{k}) = \prod_{j=1}^{3} \mathrm{P}(sd_j \mid \vec{a}_j)$$

$$P(\vec{k}) = \prod_{j=1}^{3} \mathrm{P}(\vec{a}_j)$$

因此式(2)重写为

$$\mathrm{P}(Q_i \mid D) = \eta \sum_{\vec{k}} P(Q_i \mid Q_{i1}, Q_{i2}, Q_{i3}) (\prod_{j=1}^{3} \mathrm{P}(Q_{ij} \mid \vec{a}_j) \mathrm{P}(sd_j \mid \vec{a}_j) \mathrm{P}(\vec{a}_j)) \tag{5}$$

$$\mathrm{P}(\mathrm{Q_{ij}} \mid \vec{a}_j) = 1 - \prod_{m=1}^{m_j} 1 - P(c_{jm} \mid \vec{a}_j) \tag{6}$$

$m_j$ 表示结构化查询的第 j 个部分查询中 $c_{jm}$ 的个数。

$$\mathrm{P}(sd_j \mid \vec{a}_j) = 1 - \prod_{m=1}^{n_j} 1 - P(t_{jm} \mid \vec{a}_j) \tag{7}$$

其中 $n_j$ 表示图 3 实例化后 $sd_j$ 中有效值的个数。

$P(c_{jm} \mid \vec{a}_j)$ 通过cos函数计算 $c_{jm}$ 和 $\vec{a}_j$ 的相似度来计算：

$$P(c_{jm} \mid \vec{a}_j) = \cos(c_{jm}, \vec{a}_j) = \frac{\sum_{c_{jm} \in T_j} w_{jm} g_m(\vec{a}_j)}{\sqrt{\sum_{c_{jm} \in T_j} w_{jm}^2}} \tag{8}$$

其中 $g_m(\vec{a}_j)$ 表示向量 $\vec{a}_j$ 的第 m 个值。

$P(t_{jm} \mid \vec{a}_j)$ 通过 cos 函数计算 $t_{jm}$ 和 $\vec{a}_j$ 的相似度来计算：

$$P(t_{jm} \mid \vec{a}_j) = \cos(t_{jm}, \vec{a}_j) = \frac{\sum_{t_{jm} \in T_j} w_{jm} g_m(\vec{a}_j)}{\sqrt{\sum_{t \in T_j} w_{jm}^2}} \tag{9}$$

式(8)和式(9)中的 $w_{jm}$ 是 XML 文档词汇树中各概念的权重，计算按照下面两种情况：

(1)词汇 $t_{jm}$ 出现在非叶结点 $sd_j$ 中的权重

$$w_{jm} = weight(t_{jm}, sd_j) = \ln(1 + tf(t_{jm}, sd_j)) * I(t_{jm}, sd_j)$$

其中 $tf(t_{jm}, sd_j)$ 为词汇 $t_{jm}$ 出现在 $sd_j$ 中的次数，$I(t_{jm}, sd_j)$ 是熵，反应词汇 $t_{jm}$ 在节点 $sd_j$ 的直接后继节点中的分布情况。

$$I(t_{jm}, sd_j) = \frac{\sum_{sub_k} tf(t_{jm}, sub_k) * \ln \frac{tf(t_{jm}, sub_k)}{tf(t_{jm}, sd_j)}}{tf(t_{jm}, sd_j) * \ln \frac{1}{N(sub)}}$$

其中 $sub_k$ 表示节点 $sd_j$ 的第 k 个后继节点，$N(sub)$ 表示 $sd_j$ 直接后继节点的个数。

(2)词汇 $t_{jm}$ 出现在叶结点 $sd_j$ 中的权重

$w_{jm} = weight(t_{jm}, sd_j) = \ln tf(t_{jm}, sd_j) * \ln \frac{N}{n_i}$，$\mathrm{P}(\vec{a}_j) = \frac{1}{2^{|\vec{a}_j|}}$，其中 $|\vec{a}_j|$ 表示 $\vec{a}_j$ 的变量个数。

$$P(Q_i \mid D) = \eta \times \frac{C^{k_i}}{C^{k_U}} \times \prod_{j=1}^{3} (1 - \prod_{m=1}^{m_j} 1 - \cos(c_{jm}, \vec{a}_j)) \times (1 - \prod_{m=1}^{n_j} 1 - \cos(t_{jm}, \vec{a}_j)) \times \frac{1}{2^{|\vec{a}_j|}} \tag{10}$$

令 $\alpha = \eta \times \prod_{j=1}^{4} \frac{1}{2^{|\vec{a}_j|}}$，式(10)简写为

$$\mathrm{P}(Q_i \mid O) = \alpha \frac{C^{k_i}}{C^{k_U}} \times \prod_{j=1}^{3} (1 - \prod_{m=1}^{m_j} 1 - \cos(c_{jm}, \vec{a}_j)) \times (1 - \prod_{m=1}^{n_j} 1 - \cos(t_{jm}, \vec{a}_j)) \tag{11}$$

其中，$n_j$ 是第 j 个结构单元的可能取值个数，$m_j$ 是 $Q_{ji}$ 的取值个数。

通过式(11)计算各结构化查询的概率，选择概率前 3 个的结构化查询提交给数据库执行(实验表明前 3 个最能代表用户的需求)。查询得到的文档可以按照目前多个文档排序模型来处理。

## 5 结束语

本文给出一种 XML 文档索引策略，在此基础上建立了基于贝叶斯网络的查询模型，可以准确、高效地检索复杂层次结构的 XML 文档，更具有通用性。该模型能很好地处理信息检索中的不确定性知识，对用户查询实现基于概念语义的信息检索。

## 参考文献


1 郭永明. XML 文档检索技术研究[D]. 太原: 太原理工大学, 2003.

2 Calado P, Silva A S, Laender A H F. A Bayesian Network Approach to Searching Web Database Through Keyword-based Queries[J]. Information Processing and Management, 2004, 40(5): 773-790.

3 Calado P, Silva A S, Vieira R C, et al. Searching Web Databases by Structuring Key-based Queries[C]. Proceedings of the 11th Conference on Information and Knowledge Management, New Orleans, LA, USA: ACM Press, 2002: 394-401.

4 Cristo M A P, Calado P P, Silveira M L, et al. Bayesian Belief Networks for IR[J]. International Journal of Approximate Reasoning, 2003, 34(2/3): 163-179.

5 宋 玲, 马 军, 郭家义. 支持 XML 信息检索的索引技术[J]. 计算机应用研究, 2005, 22(3): 31-33.