

集中式 P2P 环境下有效的 Top - K 查询

胡文江,高永兵,张 健,樊瑞民

(内蒙古科技大学 信息工程学院,内蒙古 包头 014010)

摘 要:目前大多数的 Peer - to - Peer (P2P) 系统只支持基于文件标识的搜索,用户不能根据文件的内容进行搜索。Top - k 查询被广泛地应用于搜索引擎中,获得了巨大的成功。可是,由于 P2P 系统是一个动态的、分散的系统,在 P2P 环境下进行 top - k 查询是具有挑战性的。提出了一种在集中式 P2P 系统中的基于中心文档的层次化的 top - k 查询算法。首先,采用层次化的方法实现分布式的 top - k 查询,将结果的合并和排序分散到 P2P 网络中的各个节点上,充分利用了网络中的资源。其次,将节点返回的结果录入到中心文档中,然后确定其分数上限,对节点进行选择,提高了查询效率。

关键词:P2P 网络;top - k 查询;检索;中心文档

中图分类号:TP311.5

文献标识码:A

文章编号:1673 - 629X(2009)05 - 0093 - 03

Efficient Top - K Query Processing in Central Peer - to - Peer Network

HU Wen-jiang, GAO Yong-bing, ZHANG Jian, FAN Rui-min

(Information Engineering Institute, Inner Mongolia University of Science and Technology, Baotou 014010, China)

Abstract: Most of the existing peer - to - peer (P2P) systems only support simple title - based search, and users cannot search the data based on their content. Top - k query is widely used in the search engine and gains great success. However, processing top - k query in central P2P network is very challenging because a P2P system is a dynamic and decentralized system. An efficient hierarchical top - k query processing algorithm based on central - document is proposed. First, a distributed query processing model for top - k query is proposed. It does top - k query in a hierarchical way. Ranking and merging of documents are distributed across the peers, which takes full advantage of the computing resource of the network. Next, a central - document is constructed for each peer according to the top - k results returned by the peer, and used to estimate the possible upper bound of the score for the peer. By the central - document information, the most possible peers are selected to send the query, so as to greatly improve the search efficiency.

Key words: P2P network; top - k query; search; central - document

0 引 言

Peer - to - Peer (P2P), 又称对等网络或对等计算, 是当前研究的热点。而在各类统计分析中经常会遇到对大量无序数据集统计出排名前 N 位元素的问题, 这类问题通常被称为 Top K 统计问题。top K 统计在网络流量测量和网络管理、数据挖掘和知识工程、大规模数据库, 以及各类统计分析工程中都有重要应用。目前大多数的 P2P 系统只支持基于文件标识的搜索^[1], 限制了 P2P 系统的应用。top - k 查询在 Web 搜索引擎中被广泛地使用并获得了巨大的成功。例如 Google, 每次查询, 它都返回匹配度最高的 top 10 个 Web 页面。人们希望使用 P2P 系统也能像使用 Web

搜索引擎一样方便, 当输入一个查询, 返回 top - k 个最匹配的结果。由于用户往往不关心所有返回的结果, 而只关心很小一部分的结果, 因此, 在 P2P 中进行 top - k 查询是十分有用的, 它能节约网络的带宽, 同时也提高了用户的满意度。由于 P2P 系统中不存在中心控制节点, 网络中的节点也可以随意加入或退出系统, 因此在 P2P 环境下进行 top - k 查询是十分具有挑战性的。

Pizza^[2]和 PeerDB^[3], 它们是 2 个具有代表性的 P2P 下的数据管理系统。但是它们都存在不同程度的缺点。Pizza 随着规模的扩大, 维护的工作量会很大。PeerDB 采用的是一种不精确的方法, 需要用户的干预。这里讨论了在集中式 P2P 系统中基于中心文档的层次化的 top - k 查询^[4]。首先, 采用层次化的方法实现分布式的 top - k 查询。利用向量空间模型 (vector space model, VSM)^[5], 进行本地的 top - k 查询, 然后聚集搜索到的所有节点的 top - k 结果产生最优的

收稿日期: 2008 - 09 - 04

基金项目: 内蒙古自然科学基金项目 (200711020813)

作者简介: 胡文江 (1959 -), 男, 教授, 研究方向为网络技术、数据库。

top - k 结果。由于 P2P 中每个查询过程都可以用一棵查询树来表示,可以分层进行 top - k 查询。儿子节点返回 top - k 结果给父亲节点,父亲节点聚集所有儿子节点的 top - k 结果和本地的 top - k 结果产生最优的 top - k 结果。分层的 top - k 查询将结果的排序和合并分散到网络中的各个节点,充分利用了网络中的计算资源。由于结果自底向上进行合并,只取合并后的 top - k 个结果向上传递,一些不相关或者较差的结果就在合并的过程中被过滤掉,因此节省了网络带宽。其次,根据各个 peer 的结果计算出各个 peer 点的分数上限。将其结果记录在中心节点的中心文档中,然后根据记录进行排序依次选出分数最好的 peer。假如当前返回的 top - k 文档的最小分数大于等于余下的 peer 的分数上限,就说明余下的 peer 不包含真正的 top - k 结果,停止排序选取,得到了最优的 top - k 结果。

1 分层的 top - k 查询

P2P 网络的搜索是一种广播式的搜索,节点除了进行本地搜索还将查询被广播发送给它的所有邻居节点。搜索范围由查询跳数 TTL 给出,查询每转发一次,TTL 减 1,当 TTL = 0 时,就停止传播查询。为了防止回路的产生,每个查询消息都有一个唯一的编号。如果节点先前已经收到查询,则说明发生了回路,节点不再继续传播查询。可以把这一查询过程用一棵查询树来表示,如图 1 所示。图中节点 a 发起查询,为查询树的根节点。查询被广播发送给节点的所有邻居节点,每次转发 TTL 减 1,当 TTL = 0 或者没有邻居的节点为叶子节点。

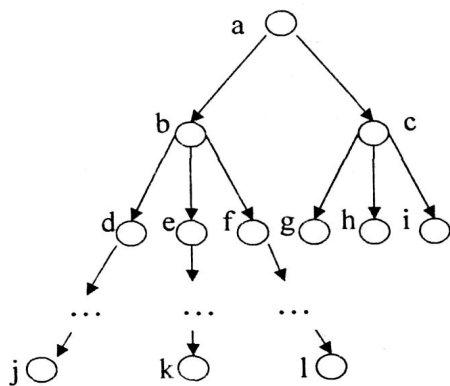


图 1 查询树

先来定义查询树。假设节点 P 为查询 Q 的发起节点,TTL 为查询跳数,则查询树 QueryTree(P, Q, TTL) 可以递归定义为:

- (1) 节点 P 为查询树 QueryTree(P, Q, TTL) 的根节点;
- (2) 假如 TTL = 0 或者节点 P 没有邻居节点(不

包含节点 P 的父亲节点),则查询树 QueryTree(P, Q, TTL) 只包含一个节点 P;

(3) 否则,假设节点 p_1, p_2, \dots, p_n 为节点 P 的邻居节点(不包含节点 P 的父亲节点),查询 Q 被广播发送到节点 p_1, p_2, \dots, p_n 。QueryTree($p_1, Q, TTL - 1$), QueryTree($p_2, Q, TTL - 1$), ..., QueryTree($p_n, Q, TTL - 1$) 为查询树 QueryTree(P, Q, TTL) 的子树,它们的根节点分别为 p_1, p_2, \dots, p_n 。节点 P 为 p_1, p_2, \dots, p_n 的父亲节点。为了防止回路的产生,假如节点 p_i (1 ≤ i ≤ n) 先前已经收到过查询 Q,则删除节点 P 到节点 p_i 的回路。

定义了查询树,定义 P2P 下的 top - k 查询为搜索查询树中与查询最匹配的 top k 个结果,top - k 查询就是搜索与查询最匹配的 top k 个文档。将查询树中所有节点的全部文档作为一个文档集,top - k 查询可以看成是搜索文档集中与查询最为相似的 top k 个文档。

利用空间向量模型(VSM)进行 top - k 查询,top - k 函数采用文档和查询的相似度函数。根据相似度函数给文档打分,取分数最高的前 k 个文档作为 top - k 结果返回。文档和查询的相似度通过计算文档向量和查询向量的夹角余弦进行衡量,具体公式为:

$$\text{Sim}(d_j, q) = \frac{d_j \cdot q}{|d_j| \times |q|} = \frac{\sum_{i=1}^n w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^n w_{i,j}^2} \times \sqrt{\sum_{i=1}^n w_{i,q}^2}}$$

其中, d_j, q 分别为文档 d_j 和查询 q 的向量表示, $d_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j})$, $q = (w_{1,q}, w_{2,q}, \dots, w_{n,q})$, $w_{n,j}$ 和 $w_{i,q}$ 分别对应词 t_i 在文档 d_j 和查询 q 中的权重。向量之间的夹角余弦通过计算向量的点积得到。

文档中词的权重一般采用 TF * IDF 规则产生, TF(term frequency) 表示词频, IDF (inverse document frequency) 表示逆文档频率。具体计算公式为:

$$w_{i,j} = \frac{f_{i,j} \times \log(N/n_i)}{\sqrt{\sum_{i=1}^n f_{i,j}^2 \times \log(N/n_i)^2}}$$

其中, $w_{i,j}$ 为词 t_i 在文档 d_j 中的权重; $f_{i,j}$ 为词 t_i 在文档 d_j 中的词频; N 为文档总数; n_i 为文档集中出现 t_i 的文档数;分母为归一化因子即将 $w_{i,j}$ 做归一化处理,使其值在 [0, 1]。

查询向量的计算相对简单,查询中关键字的权重要么是 1,要么是 0,如果该关键字在查询中出现,则为 1,否则为 0。同样,也对查询向量进行归一化处理。

声明完理论之后,可以看出最现实的做法是各个节点根据自己的本地信息进行 top - k 查询,然后聚集所有节点的 top - k 结果产生最优的 top - k 结果。将网络中的每个节点看成是一个搜索引擎,节点利用 VSM 进行本地的 top - k 查询。这即是分层的 top - k

查询。

定义 P2P 环境下的 top - k 函数为:

$$\text{Top - k}(P, Q, \text{TTL}) = \max_k(\{ \text{Local - Top - k}(p_i, Q) / p_i \mid \text{QueryTree}(P, Q, \text{TTL}) \})$$

\max_k 表示聚集查询树中所有节点的 top - k 结果,取其分数最大的 k 个文档作为最终的 top - k 结果返回。

2 利用中心文档进行 peer 的选择

层次化的广播式遍历查找^[6]是十分低效的,人们希望通过某种方法可以事先判断出那些不包含真正 top - k 结果的节点,从而提高检索的效率。由于本系统是集中式 P2P 系统,这样各个 peer 返回的 top - k 结果就可以录入到中心节点中的一个文档上去,称之为中心文档。这样就可以方便地得到各个 peer 的分数上限。这样对 peer 进行选择,剪掉无用分枝,以提高检索的效率。

2.1 peer 的分数上限

假设文档 d_j 和查询 q 的向量都经过归一化处理,

则相似度函数 $\text{Sim}(d_j, q) = \sum_{i=1}^n w_{i,j} \times w_{i,q}$ 是单调递增的,即给定两个文档 d_1 和 d_2 , $d_1 = (w_{1,1}, w_{2,1}, \dots, w_{n,1})$, $d_2 = (w_{1,2}, w_{2,2}, \dots, w_{n,2})$ 。假如对所有的 $w_{i,q}$ 0, 都有 $w_{i,1} \geq w_{i,2}$, 则 $\text{Sim}(d_1, q) \geq \text{Sim}(d_2, q)$ 。

根据上述性质查询 q 在节点 A 中出现的关键词 t_i , $w_{i,q} > 0$, 设 t_i 在 top - k 文档中的权重上限为 $w_{i,A,\text{TTL}}$, 令 $w_{i,A,\text{TTL}} = \max_{j \in \text{top - k}} w_{i,j}$ 。则又令 A 点相对于查 q 的分数上限为 $\text{UpperScore}_{q,A,\text{TTL}} = \sum_{w_{i,q} > 0} w_{i,A,\text{TTL}} \times w_{i,q}$ 。依此类推,系统中的各个 peer 均可得到其相应的分数上限。并录入到中心文档中。

2.2 基于中心文档的 peer 的选择

得到了 peer 的分数上限,而且各个点都集中在中心文档中,这样可以很容易地对 peer 进行选择。首先根据上限进行排序,选取最大分数发送给查询,如此进行。当前返回的 top - k 文档的最小分数大于等于余下的 peer 的分数上限,就说明余下的 peer 不包含真正的 top - k 结果,就停止向查询发送,得到最优 top - k 结果。

可以看出,此种方法避免了将查询广播发送到所有的邻居节点,而是根据 peer 可能的分数上限对 peer 进行选择,依次发送查询。当节点的分数上限小于等于当前 top - k 文档的最小分数时,以该节点为根的子树不包含真正的 top - k 结果,这些节点就被过滤掉,从而提高了搜索的效率。由于 peer 返回的结果都会录入到中心文档中,因此开始执行时每个节点都保证

被访问到,相当于是广播的发送查询。随着查询的执行,中心文档将会根据节点返回的 top - k 结果不断更新,利用分数上限,筛掉一些无用的节点,提高了搜索的效率。

2.3 节点的加入和退出

P2P 网络的一个重要特性就是网络是动态的,节点可以自由加入和退出网络。在 P2P 的环境下进行 top - k 查询必须考虑节点的动态加入和退出。

对新加入网络的节点 A,由于不存在该节点在中心文档上的信息,节点 A 相对于查询 q 的分数上限为

$$\text{UpperScore}_{q,A} = \sum_{i=1}^n 1 \times w_{i,q}, \text{UpperScore}_{q,A} \leq 1。$$

由于文档的分数在 0 和 1 之间,就一定保证访问节点 A,再根据节点 A 返回的 top - k 结果,更新节点 A 到中心文档中去。对离开网络的节点 A,继续搜索其他节点,直到满足查询终止条件为止。例如:图 1 中,假如节点 e 退出网络,节点 d 发现节点 e 没有响应,将把查询发送给节点 f(前提是当前 top - k 结果的最小文档分数小于节点 f 的分数上限),继续搜索。由于每个儿子节点(邻居节点)都在中心文档中有记录,儿子节点退出,父亲节点可以继续保留儿子节点在中心文档中的信息或者隔一段时间后将儿子节点的信息删除,更新操作十分简单。

3 结束语

讨论了如何在 P2P 环境下进行有效的 top - k 查询,检索与查询最为相关的 top - k 个文档。并提出了一种有效的 top - k 算法。文献[7]提出了一种邻居节点自配置的方法,根据节点返回的结果对节点的邻居进行动态配置,将那些返回结果最多的节点调整成为查询节点的直接邻居,从而更快地找到结果。研究如何利用返回的 top - k 结果调整邻居节点,进一步提高 top - k 查询的效果和效率。

参考文献:

- [1] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet applications [J]. ACM SIGCOMM Computer Communication Review, 2001, 31(4): 149 - 160.
- [2] Halevy A, Ives Z, Suciu D, et al. Schema mediation in peer data management systems[C]// In: Proc. of the 19th ICDE. Bangalore: [s. n.], 2003: 505 - 518.
- [3] Ng W S, Ooi B C, Tan K L, et al. PeerDB: A P2P-based system for distributed data sharing[C]// In: Proc. of the 19th ICDE. Bangalore: [s. n.], 2003: 633 - 644.

(下转第 98 页)

合 $\{y_1, y_2, \dots, y_n\}$;

(4) 对于数据点 y_1, y_2, \dots, y_n 利用 k 均值^[10]算法聚成 k 类。

2 实验结果分析

为了验证引用谱聚类算法在文本挖掘方面效果的提高,使用一组 200 个新闻文本的数据集进行实验和测试,文档的特征提取利用 df/idf(文档频数和逆文档频数商)方法,计算文档间的余弦值,量化文档间相似度^[11],算法使用 Pentium4, 2.0 G 主频,1 G 内存的机器运行环境,把这种处理结果和 K-均值^[12],CURE 法^[13]进行了对比。对比结果见表 1。

表 1 文本挖掘中几种聚类算法的比较

算法名称	算法效率	聚类形状	对先验知识的要求	对文本噪声的敏感性	文档顺序	处理高维数据的能力	查全率	查准率
谱聚类	$O(n \log n)$	任意	大:需要实现输入参数 k	不敏感	不敏感	高	82.5 %	83.5 %
K-均值	$O(knt)$	凸状,球状	大:需要实现输入参数 k	敏感	敏感	一般	75.6 %	85.6 %
CRUE 法	$O(n)$	任意	小	不太敏感	敏感	低	70.6 %	76.8 %

从表 1 可以看出,无论是文中使用的谱聚类方法,还是 K-均值、基于密度的方法,其运算复杂度都非常高,很难胜任海量数据的分析和处理。但是,谱聚类的使用尽管效率上有待提高,在处理高维有噪声的文本方面,其查准率和查全率却得了较好的结果。

影响算法的效率和性能是多方面的。首先,在文本的特征提取方面,把文本特征孤立看待,割裂了文本特征间的联系,影响查准率和查全率,更好的方法,应该在特征的提取和表示方面,更多地引入语义。然后,引入了 Laplacian 矩阵的计算,使得系统计算的时间和空间资源开销很大,可以在算法的计算方面进行提高。最后,文本对象间是一种高维的异构关系,我们所使用的谱聚类算法一般都把文档认为是一种同构关系,即假设所有的对象都分布在相同的欧式空间,并且对象之间只存在单一的“相似”关系,这点和现实情况很不相符。

考虑到文本挖掘是个处理大量数据的复杂问题,

文本的特征维数很高,文本的特征分布不均匀,文中的谱聚类算法要计算矩阵特征,从某种程度上影响算法的效率,因此,研究谱聚类算法的高效处理是未来的主要研究方向。另外,找到一种更加合理的比较文档间相似度的方法也是研究的重点。

参考文献:

- [1] Donth W E, Hoffman A J. Lower bounds for the partitioning of graphs[J]. IBM J Res Develop, 1973, 17: 420 - 425.
- [2] Fiedler M. Algebraic connectivity of graphs[J]. Czech Math J, 1973, 23: 298 - 305.
- [3] Hagen L, Kahng A B. New spectral methods for ratio cut partitioning and clustering[J]. IEEE Transactions on Computer-Aided Design, 1992, 11(9): 1047 - 1085.
- [4] Mohar B. Some applications of Laplace eigenvalues of graphs [C]// In: Hahn G eds. Graph Symmetry: Algebraic Methods and Applications, Vol 497 of NATO ASI Series C. [s. l.]: Kluwer, 1997: 225 - 275.
- [5] 卞月华, 吴建专, 顾国华, 等. 图论及其应用[M]. 南京: 东南大学出版社, 2002.
- [6] 周志华 王 钰. 机器学习及其应用[M]. 北京: 清华大学出版社, 2006.
- [7] Stoer M, Wanger. F. A simple min-cut algorithm[J]. Journal of the ACM, 1997, 44(4): 585 - 591.
- [8] Shi J B, Malik J. Normalized cuts and image segmentation[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888 - 905.
- [9] 苏育才, 姜翠波, 张跃辉. 矩阵理论[M]. 北京: 科学出版社, 2006.
- [10] 刘泉凤, 陆 磊. 文本挖掘中聚类算法的比较研究[J]. 计算机时代, 2005, 12: 28 - 30.
- [11] 王丽坤, 王 宏, 陆玉昌. 文本挖掘及其关键技术和方法[J]. 计算机科学, 2002, 29: 21 - 24.
- [12] Krishma K, Murty M N. Genetic K-Means Algorithm[J]. IEEE Transaction on System, Man, and Cybernetics, Part B, 1999, 29(3): 433 - 439.
- [13] Guha S, Rastogi R, Shim K. CURE: An Efficient Clustering algorithm for Large Databases [C]// Proc. 1998 ACM SIGMOD Int. Conf. Management of Data (SIGMOD '98). [s. l.]: [s. n.], 1998.

(上接第 95 页)

- [4] Halotia G, Hulgeri A, Nakhey C, et al. Keyword searching and browsing in databases using BANKS[C]// In: Proc. of the 18th ICDE. San Jose: IEEE Press, 2002: 431 - 440.
- [5] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval[M]. Boston: Addison Wesley, 1999: 27 - 30.
- [6] 张 炜, 李建中, 高 宏, 等. 基于 Peer-to-Peer 的多媒体

数据库 K-NN 查询处理[J]. 计算机科学, 2002, 29(8): 190 - 193.

- [7] Ng W, Ooi B C, Tan K L, et al. A novel peer-to-peer system based on self-configuration[J]. Journal of Software, 2003, 14(2): 237 - 246.