基于谣言传播及兴趣挖掘的搜索算法①

陈　佳[1,2]　梁克维[2]　(1.香港理工大学 应用数学系 香港 九龙;
2.浙江大学 理学院 浙江 杭州 310027)

摘　要： P2P 常用的搜索及传播算法占用大量带宽，阻碍了信息流通的顺畅。以占用带宽较少的谣言传播算法为基础，结合了兴趣挖掘算法，提出了导向性强而且带宽占用率低的搜索算法。并对该算法进行了软件简化建模模拟以及测试验证。

关键词： 谣言传播 兴趣挖掘 概率播放机制 转发策略

Searching Algorithm Based on Rumor Spreading and Interest Mining

CHEN Jia[1,2], LIANG Ke-Wei[2]
(1.The Hong Kong Polytechnic University, Kowloon, Hong Kong, China)
2.Zhejiang University, Hangzhou 310027, China)

Abstract: It is true that the commonly-used P2P searching and spreading algorithm occupies a large bandwidth so that the smooth flow of information is hindered. In this paper, based on the rumor spreading algorithm, combining with the core thinking of interest mining technology, a searching algorithm has been introduced, which has both the advantage of low range of bandwidth and the advantage of strong directing orientation. Furthermore, the algorithm has been simulated, tested, demonstrated and evaluated.

Keywords: rumors spreading; interest mining; probability broadcast mechanism; transmitting strategy

1　引言

近年来，P2P(对等网络)应用非常广泛，其中最流行的应用是非结构化 P2P 环境下的文件共享。P2P 改进的主要目标是：找到更好的搜索机制，使得系统既有好的检索结果，又有很高的搜索效率。然而，现有的非结构化 P2P 系统都不能很好的同时满足这两个目标。目前，大量的 P2P 应用基于简单的 Flooding 广播机制，在这类系统中，任意节点将接收到的查询请求转发给其所有的邻居节点。因此，请求消息数量被指数级放大。在当前 Internet 主干网络的通信开销中，有 40％以上被 P2P 应用消耗，而这种情况随着更多大规模 P2P 系统的出现而变得更加严重。由此可知，一个好的搜索算法是减少通信开销的关键。

在对谣言传播的研究中可以得出以下事实：人类个体传播谣言的方式是从自己的邻居(好友、同僚等)每次随机选择一个个体进行传播。同时，一个有趣的现象是：忽略个人的好恶因素，人类个体传播某个谣言的兴趣会随着多次收到同样的谣言而降低，即当多次听说某一谣言时，个体会本能地认为该谣言已经广泛传播，从而停止自身的传播，因此可认为这种“兴趣衰减”有利于减少谣言在人际网络中传播的开销浪费。另一个隐含的事实是，谣言传播所基于的人际网络具有高度的聚合性，同样认为，聚合性保证了谣言传播具有较高的覆盖度[1]。由此产生了基于谣言传播

① 基金项目:浙江大学优秀青年教师资助紫金计划(107100-811139)
收稿时间:2009-04-23

机制的搜索算法，其核心内容是单一节点根据收到消息次数决定再度转发该消息的概率，此概率随着接收消息次数增多而迅速减小，达到减少重复发送的目的。

另一种优化算法被称为兴趣挖掘算法，其核心采用向量空间模型(VSM)方法[2,3]，将文件资源利用关键词出现频率向量化，然后进行聚类[4]，形成兴趣簇，同为向量形式。所有兴趣簇维数相同，以便进行相似度判定[5]。每个节点都存储其相邻节点的兴趣簇，以之代表资源信息。而为了方便进行匹配，查询请求也扩充为与兴趣簇相同的维数，具体方法是若该维度代表关键词在请求中出现，则该维度为 1，否则为 0。之后根据相似度大小确定消息发送的先后,从而达到加快资源反馈速度的目的。

然而对谣言算法来说，在传播过程中是完全随机而且不分发送先后的，比起 Flooding 算法的优势仅在于减少一部分节点传播，而这部分优势是以损失查全率为代价的，若从某个邻节点偏少的节点发起查询极有可能无法搜到所需资源。而对于兴趣挖掘算法而言，其搜索的总量是与 Flooding 算法相同的，尽管这保证了查全率毫无损失，但是其兴趣指向仅能保证较快地搜索到高契合度的资源而无法减少网络开销。同时我们发现这两种算法的优势有着很好的互补特征，因此结合谣言传播机制(Rumor Spreading Mechanism)及兴趣挖掘算法(Interest Mining)，产生了一种新算法，即基于谣言传播及兴趣挖掘的非结构化 P2P 搜索算法，简称 RSIM(Rumor Spreading and Interest Mining)算法。本文介绍了该算法的原理及核心机制，并对其进行了软件简化建模模拟以及测试验证。

2 RSIM算法

RSIM 算法以占用带宽较少的谣言传播算法为基础并针对其盲目性融入兴趣挖掘算法的导向性优势。具体来说，其优势体现在三方面：(1)增加指向性可以保证不传播的一部分节点与查询内容关系很小，从而减少查全率损失。(2)根据邻居资源信息与查询内容相关性决定传播次序先后，这在实际中意味着较好的结果将会较早地反馈。(3)通过在传播双方发送列表中将对方剔除的方式减少请求回流的现象。

与兴趣挖掘算法相同，RSIM 算法需要进行对节点进行相同的文本特征提取的预处理，使得计算优先级及匹配计算时都采用相同维度向量。我们利用兴趣索引表来存储邻居节点兴趣，兴趣索引表的结构为<Interest，Address>，其中 Interest 项表示兴趣，address 表示该兴趣所在的邻居节点 IP 地址。节点加入到网络的同时，将自己的兴趣特征向量发送到邻居节点，该邻居节点接受到这个信息后，存入自己的兴趣索引表中。接收到查询请求后，将查询请求与兴趣索引表中的所有兴趣进行相似度计算：

$$sim(q,c_{ij})=\frac{\sum_{k=1}^{n} q_k \times w_{jk}}{\sqrt{\left(\sum_{k=1}^{n} q_k^2\right)\times\left(\sum_{k=1}^{n} w_{jk}^2\right)}} \tag{1}$$

其中，q_k 代表查询请求 q 中关键词 k 的权重，c_{ij} 表示节点 i 的第 j 个兴趣，w_{jk} 表示节点 i 第 j 个兴趣中关键词 k 的权重。故公式(1)其实就是查询请求向量与兴趣向量间的余弦。该值越大，说明 q,c_{ij} 的相似程度越高。同样，这里要求查询向量与兴趣向量维数相同。

利用相似度，我们来确定播放节点的优先级。由于任一节点都有若干兴趣，而每个节点重要程度都是相同的，因此我们将优先级定义为节点内所有兴趣与查询请求相似度的平均值，即：

$$R(P_i,q)=\left(\sum_{j=1}^{C_i} sim(c_j,q)\right)\Big/ C_i \tag{2}$$

式中，P_i 表示节点 i，C_i 表示节点 i 的兴趣个数，c_j 表示第 j 个兴趣。$R(P_i,q)$ 表示该节点发送请求时的优先级，该值越大，意味着节点 i 越有可能存有查询 q 所需信息。

在谣言传播算法中[2]，任意节点在收到消息 M 时，根据概率 P 决定是否向其邻居节点 u 传递消息 M，其中 $u\in N_v-S_v-R_v$。N_v 为节点 v 邻集，S 为 v 接受过信息的节点集，R_v 为 v 已发送节点集。p_v 定义为：

$$p_v=1-\left(1/h\right)^{|N_v|-|S_v|} \tag{3}$$

其中，h 为某一常量系数，且 h>1，$|N_v|$，$|S_v|$分别表示 N_v 和 S_v 中的元素个数。

该算法的主要问题在于转发时的随机性。该算法由于只有一定概率发送给 N_v-S_v 中的节点，从而与洪

泛算法相比大幅减少了冗余信息。但是，该算法在决定转发顺序时是任意的，同时，在转发过程中，其转发概率 p_v 恒定不变。这使得在转发节点选择上缺乏针对性，因此该算法的查全率并不理想，其冗余消息的减少是以大量牺牲查询成功率为代价的。

为了提高查询成功率，我们强化了节点转发时的目的性。为此，我们加入相似度及优先级；同时，为了在发送概率中体现节点的兴趣差异，需要对优先级进行改造。我们将节点 v 的邻居节点按优先级进行排序，优先级最高的序号为 1，次高的序号为 2，依此类推，则可将(3)改进为：

$$p_{va}=1-\left(1/h\right)^{|N_v|+|S_v|-O_a+1} \quad (4)$$

其中，p_{va} 表示节点 v 向其邻居节点 a 发送消息的概率，O_a 表示邻居节点 a 的序号。这样，当 a 优先级为最高时，$p_{va}=p_v$；随着 a 的序号增大，p_{va} 迅速减小，从而使得消息传播以节点兴趣为指导，达到提高查询成功率的目的。

但是，公式(3)与公式(4)有一个共同的问题，当 $|N_v|$ 本身很小时，即使是首次发送消息，相比于度数(邻节点个数)更高的节点，该节点的转发概率很小。这将会使度数较低的节点有很大几率产生“断链”情况，即当消息传递到此处时，由于控制是否发送消息的概率很小使得消息有很大几率无法继续发送。当这类节点作为消息始发节点时情况更为明显，几乎无法开始查询。为避免这种情况，我们对公式进行进一步改进，对度数不同节点产生的指数加以区别。我们决定用相对差值代替绝对差值来作为 $1/h$ 的指数。从而，公式改为：

$$p_{va}=1-\left(1/h\right)^{\frac{|N_v|+|S_v|-O_a+1}{|N_v|}}=1-\left(1/h\right)^{1-\frac{|S_v|+O_a-1}{|N_v|}} \quad (5)$$

在这种情况下，$1/h$ 幂的增减性不变，但其值域定在了 $[0,1]$ 之间，已经无关于节点 v 的邻居节点数 $|N_v|$ 了，因而即使 v 的度数很低，p_{va} 的值仍可以接近于 1。同时，由于 $1/h$ 幂不超过 1，所以(5)式中 $p_{va}\le 1-\frac{1}{h}$，而在(4)式中，$p_{va}\le 1-\left(\frac{1}{h}\right)^{|N_v|}$。因此为保证两式首次发送几率相同，(5)式中的 h’要远大于(3)，(4)式中的 h。例如在原谣言传播算法中，假设节点的平均度数 $|N_v|\approx 10$ 情况下，h 最优值取 4，首次发送消息时，播放概率约为 $1-10^{-6}$。因而 h’值取 10^6 时，改进前后两算法首次发送时的播放概率相当。

3 RSIM算法的模拟实现

3.1 RSIM 模拟模型的假设与前提

为了避免无限地传播搜索请求，无限延长搜索时间，增大网络整体负担，需要对消息传播加以控制。从而，在网络中定义一个消息生存时间 TTL，每当消息经过一个节点，TTL 减一，当 TTL=0 时，就停止传播，不再发送。这样就可以将搜索的空间规模维持在一个可控范围内。

本文算法采用 VC++6.0 实现，受硬件及条件限制，无法完全按照算法要求实现。因此在某些方面进行了抽象。

① 由于文本聚类需要大量不同类型文件才能使聚类结果从宏观上趋于稳定。在无法达到此要求的情况下，我们将文件与节点兴趣直接用向量表示。同时，对网络中所有节点中的文件向量进行聚类工作量仍然很大，因此，我们认为文件资源与节点兴趣相同，即在模拟中，节点中只存放代表节点兴趣簇的向量组，在进行相似度计算时，直接用这些向量与搜索消息向量进行余弦计算，而搜索最终的查询结果同样是这些向量。

② 关于向量维数的问题。如果按照文献[3]的要求采用 30 维向量，单机运行某个节点的过程没有问题，但对于单机模拟整个网络运作就很吃力了。因此我们把兴趣簇向量维数简化至 10 维；同时，单个节点兴趣簇个数限定为 3 个。

③ 网络中的资源是多种多样的，但是经过我们的聚类可以将其简化为有限多的兴趣簇，使之构成一个集，所有的兴趣均从该兴趣集中抽取。因此在模拟中，我们随机生成一个 n×m 的矩阵来表示兴趣集。其中 n 表示兴趣集合中兴趣的个数，该矩阵每行表示一个 m 维的兴趣向量。

④ 关于查询请求资源匹配问题。按照前文介绍，查询请求向量形式中，其分量仅有 0，1 两种情况。由于我们减少了维数和资源数，如果还采用这种查询

请求向量并使用非精确匹配方法(通过判别查询请求与资源向量的相似度是否超过预定值来确定资源是否为所求的方法)，则可能会使大多数资源均符合任意查询结果，无法评判该算法的优劣。因此，我们的查询请求向量分量采用随机生成的方式，而在搜索中则采用非精确匹配方法进行判别。

3.2 RSIM 算法存储结构的实现

要对 P2P 网络进行模拟，首先必须模拟网络中存储的信息，其中包括节点内部信息与网络链接信息两部分。由于我们对节点存储信息进行了简化，使之成为向量，因此我们只需考虑对向量的存储；而对于网络链接，我们采用图论中邻接矩阵的方式进行储存。

最终我们采用了三个部分来模拟储存：

第 1 部分是兴趣矩阵。

第 2 部分是网络的邻接矩阵，其中元素除对角线外也全部随机生成，但只可为 0 或 1，对角元素全为 0。

第 3 部分是节点本身内容的储存，采用了链表结构。每个链表节点中包含 Interest1，Interest2，Interest3 存储的网络节点三个兴趣的序号，具体兴趣需在兴趣矩阵(第一部分)中查询；Neighbor 用于储存节点的相邻节点信息，该数列根据邻接矩阵(第②部分)生成。

3.3 RSIM 算法传播过程的实现

对于 P2P 搜索的模拟中另一个重要方面则是搜索消息播放策略的模拟。我们将其分为两部分进行模拟：单次消息发送的处理及资源匹配判定；全局消息传播路径及搜索结果。

不同于实际中的搜索情况，模拟中的节点不可能向若干节点同时发送消息，因而采用链表结构，处理消息时可以按照消息链表顺序对消息进行处理。消息节点结构代表这查询请求在某一方向上的信息，其中包含 Found 存储该消息查询到的资源数；TimeToLive 存储该消息被指定的生存跳数；Query 存储消息所查询的内容向量，一般情况下该向量在整个消息链中不变；Path 存储该消息已走过路径，该向量中存有消息途经的所有节点在节点链中的编号。

单次消息发送处理及资源匹配判定算法步骤为：

① 在给出的消息节点中得到已知消息传播路径的最后一点，从而确定单次发送的源节点，目标节点全局传播时给定；

② 判别目标节点中的资源是否满足请求，若是则 Found++；

③ 从发送源节点的发送节点列表中删除目标节点；

④ 从目标节点的发送节点列表中删除源节点；

⑤ 在该消息节点中的消息传播路径序列最后添加目标节点。

单次消息发送处理流程函数参数包括一个网络节点链表，目标节点号以及兴趣矩阵。在这里我们设定与请求向量相似度超过 0.9 即认为该资源即为所求。从本次发送双方节点的邻节点列中将对方节点删除可以避免消息再度经过此边，避免消息传递形成环，减轻网络负担。

全局消息传播策略及搜索结果算法的步骤为：

① 生成消息链，包括随机生成查询请求向量，给定 TTL 以及给定发起查询节点并将该节点写入路径列中；

② 对起始节点进行消息判别，看是否有满足消息请求的信息存在；

③ 将消息节点指针 pMessage 指向消息链的首节点；

④ 从 pMessage 所指路径中最后一点的邻居节点列中得到未向该节点发出也未收到来自该节点信息的邻居节点，这些点构成发送列表；

⑤ 根据发送列表中点的个数复制消息节点；

⑥ 根据查询请求与发送列表中的节点兴趣的平均相似度确定发送优先级，选出其中最高的一个节点；

⑦ 利用公式(5)根据发送列表中的剩余点数来计算发送概率，并根据概率确定是否发送，若否，则转⑨；

⑧ 以 pMessage 指向的消息节点以及 7 中得出的节点为参量完成上部分所讲的单次发送；

⑨ pMessage 指向下一消息节点，转④。

全局消息传播流程函数参量为消息传递链，兴趣矩阵和发送概率公式中的常数 h。本函数每次执行都是对消息传播的最外围节点进行处理，因此进行几次循环即为消息行进几跳。

这样，即使通过其他路径传到源节点，也会由于目标节点已从源节点的邻居节点中删除从而不会重复该次传播。从而将重复传播带来的损耗降到最低。而 TTL 的作用则可通过循环次数控制来实现。本算法采用了广度优先的思路，即先对消息源半径为 1 处的节点进行处理，然后为半径 2 处，半径 3 处，…，半径

逐渐扩大直到 TTL。该算法最终得到的结果是处理后的消息链，其中储存了消息在所有方向上所走的路径以及每条路径上找到的资源数。

4 RSIM算法的实验评测

由于本算法具有较强的导向性，因此适合采用以下标准评价性能：

① 通信总跳数与查全率：查全率=查到的资源数/网络中的总资源数。通信总跳数与查全率的关系变化能够反映算法冗余程度。

② 平均跳数：平均跳数是指搜索路径长度的平均值，平均跳数是决定等待时间的直接因素。

③ 查到一定数量结果所用跳数。

同样反映查询花费时间，由于将查询过程理想化，所以单个节点处理不花时间，查询时间耗费全部在传输过程中。从而查询请求所经边数即可作为查询请求经过时间的度量。

我们将用本算法的上述性能与现在最常用的Flooding(洪泛)算法进行比较。

为了使结果更为接近实际情况，我们采用 2000 节点规模的网络模拟测试。同时为增加查询的成功率，我们设定兴趣矩阵规模为 200×10，即 200 个 10 维备选兴趣向量。并对网络中节点的平均度数进行不同的设定。本测试共分三部分，第一部分为给定消息生存周期条件下 RSIM 算法与洪泛法的对比，第二部分为给定消息上限情况下两种算法总路径长度与响应消息量的对比，第三部分为谣言传播算法在不同的 h 取值下的搜索状况对比。在前两部分 h 均取 105。

4.1 给定消息生存周期条件下RSIM算法与Flooding法的对比

综合上节讲述的评价指标，我们将直接测量一些信息，包括：两种算法覆盖点数，找到的资源数，可供响应的资源数，途经的总跳数，网络中具有的总资源数。此外还有一些信息由我们指定，包括消息生存时间和邻居生成概率，后者乘以网络规模即为网络中节点的期望邻居数。在这些资源中，可供响应的资源数是指每条消息传播路径上找到的资源数之和，而找到资源数则是将可供响应资源数中的重复资源排除后的剩余资源数。这两者的比值可以反映结果的冗余程度，比值越高，冗余程度越低。而找到资源数与总资源数的比值则是查全率。直接测量结果如表 1 所示。

表 1　冗余程度对比

试验序号	总资源数	RSIM 算法					洪泛法				
		覆盖点数	找到资源	途经路径	响应资源	冗余度比	覆盖点数	找到资源	途经路径	响应资源	冗余度比
1	713	129	55	252	106	0.519	134	56	280	124	0.452
2	341	184	23	363	74	0.311	191	23	408	80	0.288
3	299	152	23	292	43	0.535	161	23	324	45	0.511
4	327	103	20	191	20	1	124	22	246	22	1
5	550	174	59	342	135	0.437	185	60	394	142	0.423

表 1 为消息生存周期在 2 跳时的情况，由该表我们可以发现两者的响应资源与找到资源数差别不大，RSIM 算法的覆盖点数略少，而途经路径显著少于 Flooding 法的路径。由此可知在维持找到资源数与响应资源数的情况下，RSIM 算法所占用的网络资源远小于洪泛算法。我们以两种算法的路径为横坐标，查全率为纵坐标作图，结果如图 1 所示。

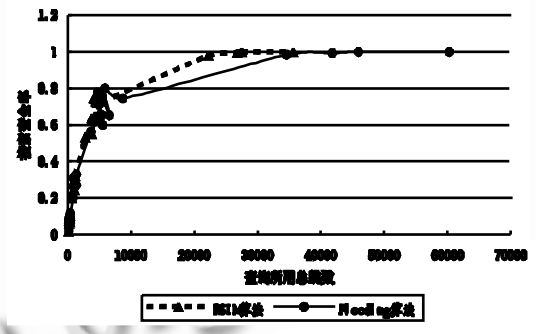

图 1　查全率 vs 总跳数

在图 1 中我们发现在查全率达到 0.8 以上后，RSIM 的谣言算法所用的总跳数要明显少于 Flooding 算法。下面我们来观察两算法的平均跳数关于网络中总资源数的变化，如图 2 所示：

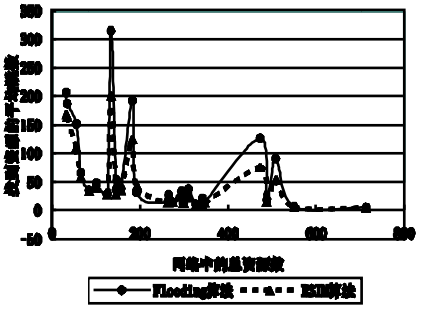

图 2　平均跳数 vs 总资源数

由图 2 中我们可以看出，在全局范围内，RSIM 算法平均跳数均小于 Flooding 算法。而在消息起始点不好(离绝大多数资源较远，在给定跳数内无法搜到大量结果，对应于图 2 中的峰值)时，两者的差别尤其明显。综合上述分析，我们认为在给定消息生存时间时，RSIM 谣言法要优于 Flooding 法。

4.2 给定消息上限情况下两种算法总路径长度对比

我们给定消息上限 20，一旦查到 20 条不同的资源记录立刻停止查找返回结果，又由于实际中用户等待时间有限，因此我们也设定了消息生存周期以免不断查询下去。在本部分我们集中考察两种算法的途经路径，因为在结果数给定的情况下途经路径所确定的用户等待时间将是评价的唯一标准。经过若干次实验得到图 3。

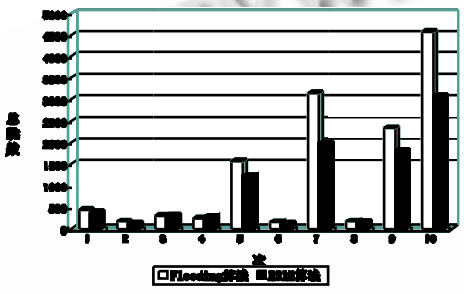

图 3 给定资源数的跳数对比

由于网络资源与搜索起始位置是随机生成的，因此资源总数及所在位置很不稳定。由图 3 可知，搜索到 20 个资源所用跳数在不同实验中差别极大；但是 RSIM 谣言算法所用跳数始终少于洪泛算法，而且总跳数越大差别越明显。因此在搜索到令用户满意数量结果的实验前提下，本文提出的 RSIM 谣言算法仍优于洪泛算法。

4.3 谣言传播算法在不同的 h 取值下的搜索状况对比

实验内容是跟踪不同的 h 取值下 RSIM 算法的表现比较，为方便起见，我们的其他参数设定与第 4.2 节实验内容类似。即同样考察在搜索结果与消息生存时间固定条件下所用总跳数。但由于结果数量随着资源总量大小而浮动，因此我们将第 4.2 节中的 20 个结果改为搜到全网络 10％结果。同时我们还设定邻居生成概率 0.005(即平均每节点生成 10 个邻居)，200 个备选 10 维兴趣向量，消息生存周期为 3。这样我们可以在尽量符合实际情况条件下，提高搜索到结果的可能。我们对在 h 分别取 1，10，102，103，104，105，106，107 等值时所取跳数值进行记录，得到如表 2 所示结果：

表 2 不同 h 值时总跳数对比

h 数量级	0	1	2	3	4	5	6	7
总跳数	793	413	457	349	439	800	164	362

表 2 中我们可以看出在 h 取值过大或过小时总跳数均偏大，而在处总跳数达到极小值。这与 2 中所述原谣言算法与 RSIM 算法首次传播发送概率相当时所取 h 值一致。因而 RSIM 算法与原谣言传播算法在最优情况下的首发概率相同，这也符合 RSIM 算法同样采取谣言传播机制作为核心的事实。

5 结语

本文以兴趣挖掘的思路改进谣言传播算法，用软件模拟了非结构化搜索平台，并在此平台上实现了算法的传播过程，还对其一些性能进行了跟踪观测，最后与该平台上实现的 Flooding 算法进行了对比评价，观察了 RSIM 算法在不同参数情形下的表现，并对参数设定给出了有参考价值的建议。

RSIM 算法在谣言算法小冗余量的基础上加强了兴趣导向性，提高了搜索目的性，最终提高了搜索成功率。从测试结果看，该算法要明显优于现在常用的算法，其代价是增加了单个节点中的空间占用，但是这些占用对于现在的计算机硬件水平来说可以忽略。因此本算法中做出的改进是具有实际意义的。

参考文献

1 李庆华,张阳,王多强.P2P 网络中基于谣言传播机制的资源搜索算法.计算机应用, 2005,259(11):2465－2467.

2 谭义红, 陈治平, 林亚平. 基于兴趣挖掘的非结构化 P2P 搜索机制研究与实现.计算机应用, 2006,26(5): 1164－1166.

3 杨颖,韩忠明,杨磊.兴趣子空间挖掘算法在高维数据聚类中的应用.计算机工程, 2007,33(2):12－17.

4 康楠,金蓓弘,李京.面向 Blog 的兴趣挖掘和推荐系统.计算机工程, 2008,34(2):72－74.

5 沈洁,胡金初.P2P 搜索新技术,智能搜索技术. 微机发展, 2005,15(11):91－93.