

P2P 网络中基于增量学习的节点聚类

张化祥¹ 刘培德² 黄上腾³

(山东师范大学计算机科学系 济南 250014)¹ (山东经济学院计算机系 济南 250014)²

(上海交通大学计算机科学与工程系 上海 200030)³

摘要 本文研究了 p2p 网络中基于内容的节点聚类。基于文件名关键词精确匹配的查询没有考虑文本语义及内容相似性。如果能够根据节点发布内容的相似性,建立节点聚类,信息查询在类内进行,必将提高查询效率。本文提出了一种基于增量学习的节点聚类方法,通过兴趣爬虫代理计算节点得分,据此判断一个节点是否可以加入节点簇。实验表明,节点簇的建立可以有效地提高 p2p 网络的查询效率。

关键词 对等网络,聚类,增量学习

Peer Clustering Based on Incremental Learning in P2P Networks

ZHANG Hua-Xiang¹ LIU Pei-De² HUANG Shang-Teng³

(Dept. of Computer Science, Shandong Normal Univ., Jinan 250014)¹

(Dept. of Computer Science, Shandong Economics Institute, Jinan 250014)²

(Dept. of Computer Science & Engineering, Shanghai Jiaotong Univ., Shanghai 200030)³

Abstract This paper discusses the content-based peer clustering in peer-to-peer networks. Information retrieval based on accurate match of keywords in filenames ignores the document semantics and the similarity between documents. If peers are clustered according to the similarity between their released documents of a special interest topic, and the information query is executed among peers of a specific cluster, the efficiency should be improved. We propose an incremental learning approach to peer clustering, and employ an interest crawler agent to calculate a peer's score. Whether a peer joins in a cluster or not is determined by its score. Experimental results demonstrate that clustering of peers in hybrid p2p networks is both accurate and more efficient for information retrieval.

Keywords Peer-to-peer networks, Cluster, Incremental learning

1 引言

对等计算(peer-to peer computing, p2p)中节点完全对等,节点同时扮演着数据、服务提供者 and 接受者,并具有自治性,可独立于系统中其他节点运行。节点间具有自组织性,节点可根据一定的规则自动选择感兴趣的节点建立不同的兴趣类。p2p 同时具有很好的可扩展性,新的节点可以随时网络,并带来新的信息和服务资源,提高系统的运算和服务能力;系统不存在单点出错和访问瓶颈问题,文件的访问分布在整个网络上,而不集中在一个服务器上,不会出现单点访问过载;同时,p2p 网络可以充分利用网络中空闲的系统软、硬件资源,提高系统的资源利用率,并能满足不同的应用需求。目前 p2p 的研究主要集中在系统体系结构及算法、信息发布与查询及查询动态优化等方面。而信息资源的发布与查询在 p2p 的研究中占有重要地位。本文研究对等计算模式下,信息资源的发布与查询的机制。

p2p 计算模式下,资源查询与系统体系结构相关。Naspter^[1]基于混合型网络体系结构,系统通过一个或多个专用服务器管理网络资源元数据。当节点给出查询主题词后,查询消息被发送到服务器。服务器根据存储的节点元数据,查询出存储相关主题节点的 IP 地址,并将查询消息发送到提交节点,节点根据 IP 地址找到对应节点并实施下载。该模式通过专门服务器管理元数据,增加系统开销,同时存在计算瓶

颈和单点出错风险,又基于主题词查询,只能实施精确匹配。Gnutella^[2]以完全非结构化的体系结构为基础实施信息的发布和查询,系统中没有集中式管理协议,资源以主题词的形式由不同用户在各自的节点上发布。用户要查询感兴趣的主体,首先查询本地共享文件库,同时将查询消息通过泛洪(Flooding)广播到相邻的节点。接到消息的节点查询本地文件库的同时,以同样的方式广播消息。消息每转发一次,其生命周期 TTL(Time-to-live)减 1,直至 TTL 为 0。基于该模式的信息查询,消息在网络中以指数方式增加,消息发送盲目,无针对性,造成带宽浪费,同时给非信息目标节点造成很多不必要的计算。因为接到消息的节点都要进行主题词比较,确定是否含有要查询的文件资源。当系统资源均匀分布于不同节点时,增加新的含有主题词的节点并不会带来系统效率的显著提高。该模式下的查询效率极其低下。文[3]提出了提高非结构模式下的信息查询效率的方法,但不能从根本上解决问题。Triantafillou^[4]提出节点簇的概念,通过预先定制节点簇,加速资源查询,问题是直接建立节点簇忽略了 p2p 系统的动态性。Past^[5]等都是基于结构化体系结构的文件共享系统,利用分布式哈希表将要发布的文件直接映射到与其文件名哈希表值相同或相近的哈希表值所标识的节点上。路由算法 Can^[6]和 Chord^[7]等都采用哈希表路由规则,信息查询可以很明确地确定目标节点。该类应用路由效率高,但由于文件通过哈希表值标识,节点只存储与自身标识符哈希值相同

张化祥 博士,副教授,主要从事人工智能、机器学习、数据挖掘和分类器设计;黄上腾 教授,博导,主要从事 CIMS、数据库及数据挖掘技术研究。

或相近的文件,限制了 p2p 系统的自治性和自组织性,失去了对等计算的优势。目前大多数信息查询基于文件名中的项或关键词匹配,基于文件内容的查询开始成为新的研究热点。文[8]提出一种基于内容的信息查询策略,通过计算叶节点文本集合内容与查询模型间的 K-L 分歧(K-L divergence),对叶节点进行排序,并将查询消息发送到排在前面的数个叶节点。方法要求目录节点对其所有叶节点计算 K-L 分歧,计算无针对性,效率低,同时目录节点间以泛洪传递信息,增加系统负载。

基于以上分析,我们认为现有的 p2p 系统信息发布及资源定位机制存在一定的局限性。为此提出一种基于内容的信息节点资源聚类 and 查询机制。我们建立基于混合型结构 p2p 系统的信息查询机制,提出了一个基于增量学习的兴趣爬虫代理。基本思想是爬虫代理根据其所属节点的文本特征,计算相同兴趣类节点文本间的相似度,并通过增量学习的方式评估爬行的节点。爬虫属主节点据此建立节点簇,并对节点簇内的节点进行排序,加快信息的查询。通过建立具有相似内容的节点聚类,提高信息查询的在线命中率和效率。

2 相关工作

结构化的 p2p 系统虽有较高的路由效率,但忽略了对等节点的自治性和自组织性,文件严格按照哈希规则存储在相应的节点上。依据主题词查询时,可以实现精确匹配。但当文件的发布者与查询者对相同的信息所关注的主题词存在偏差时,查询往往失败,更多的查询不能通过精确匹配实现,需要基于一定的文本语义实现查询。同样,完全非结构化的 p2p 系统,存在盲目的消息泛洪,浪费系统计算资源和网络资源,查询的效率还低。目前更多地是将两者结合^[9],因而出现了混合结构的 p2p 网络。消息路由通过将结构化的路由算法和非结构化的路由算法结合实现。本文基于混合结构的实验平台,验证算法的效率等指标。

p2p 网络中的信息索引有本地索引、集中索引和分布式索引三种^[10]。Naspter 通过集中索引信息实施查询路由,存在单点错误风险。改进后的 Guntella^[11]将节点分成叶节点和目录节点两类。叶节点生成发布文件标示的描述,目录节点根据这些描述选择叶节点,并向其发送查询消息。该类应用只能通过对文件名中项的匹配实现信息查询,并且目录节点间仍然采用泛洪发送查询消息。JXTA^[12]是基于混合 p2p 体系结构进行信息查询的又一实例。目录节点根据叶节点登记发布信息的相关内容路由查询消息,实现了目录节点向目录节点及叶节点路由查询消息的算法。叶节点登记的发布信息的内容直接影响着信息查询。文[13]研究了非结构化体系结构 p2p 网络基于内容的信息查询。系统中的每个节点收集其它节点的索引信息,并根据收集到的信息,应用 tf/idf 算法决定要转发查询消息的节点。可以看出动态环境下,节点收集并存储其它所有节点发布内容的相关信息的效率不高。为提高信息查询效率,查询路由算法^[14]利用邻居节点的历史查询信息来影响节点查询信息的路由,算法针对临近点的查询效率得到提高,但查询远处节点的内容比较困难。

3 信息管理与发布

网络拓扑结构影响信息索引,同时影响到信息查询路由算法。借助结构化信息查询的高效性,又不失 p2p 网络的非结构化特点,我们在网络中设立两类节点:一类用于存储节点

索引信息,称为服务节点 SP(Service Peer);另一类为存储文件的普通节点,称为数据节点 DP(Data Peer)。所有 SP 节点通过结构化协议连接,并以非结构化协议与 DP 节点连接,所有 DP 节点以非结构化协议连接。相邻 SP 节点间相互备份数据,减少单点错误。

采用这种拓扑结构,目的是提高系统的查询效率,同时扩展精确匹配查询模式,使其适合基于内容和语义的查询。DP 间的非结构协议连接,保留系统的动态性和可扩展性。节点的加入与离开不会对系统带来大的影响。同时文件存储在不同 DP 节点上,方便维护。

我们知道每个 DP 节点可能存储不同类的多个文件,这是由节点用户的兴趣所决定的。同时节点所提交的查询消息不一定是该节点用户所存储文件的类的同类信息。以前的研究大多基于查询消息提交者的集中于某些类的假设,而实际上提交的查询消息多种多样。基于用户可提交任意查询消息的假设,我们建立新的信息索引和查询机制。

按照科学分类法,将节点发布信息分成多个不同的大类。根据类关键字的哈希值,将 DP 节点的(ip、类信息、类的关键词)信息发布到具有相同或相近服务器名哈希值的 SP 上。此时,不同的 SP 管理含有不同类信息的 DP 节点元数据,同时,每个 DP 节点根据其发布共享信息的种类数,同时在多个 SP 上发布元数据信息。当新的节点加入系统时,该节点主动将含有该类信息的元数据上传到相应的 SP。SP 定时更新元数据记录,通过向 DP 节点发送代理的方式,收集属于其管理类的信息发布节点上该类的文件总数、关键词及词频等信息。该代理在 DP 端运行,减少 SP 的负载,同时一旦发现新的节点加入,就将新加入节点的相关信息登记到 SP 上;若发现有节点离开,就将其登记的信息转入到停止服务一类信息,一旦该节点再次加入网络,就从停止服务信息中将其信息激活。SP 通过移动代理,收集 DP 节点发布信息的相关元数据。

4 节点聚类

节点聚类的目的是将在 SP 上登记的节点按照它们发布文件内容间的相似性进行再次分类。当节点空闲时,节点根据不同兴趣所属类,向对应的 SP 提交查询消息,如计算机技术中的对等计算。首先确定对等计算属于计算机技术类,而计算机技术类对应的 SP 接受节点发来的查询消息,检索所存储的节点元数据,并将相关节点的基本信息发送给查询提交节点。查询提交节点根据接收的信息判断哪些节点已经成为其对等计算的类成员,哪些节点还没有成为其对等计算的类成员。此时,查询提交节点尚未加入该兴趣节点簇的节点发送兴趣爬虫代理,计算文档的相似度,根据阈值决定是否将其加入到该簇中,并对节点按相似度排序。

4.1 基于增量学习的相似度计算

节点刚加入 p2p 网络时,还没有建立节点簇,此时需要一定的算法建立其节点簇。我们可以借鉴主题驱动下的 Web 资源的发现机制来解决这个问题。Web 爬虫用于信息收集,它沿着已爬行页面中的超链接来遍历 Web,并下载相关 Web 页面,收集 Web 页面^[15]。一般 Web 爬虫通常用于搜索引擎中,作为搜索引擎的页面收集系统。它通常以宽度优先的模式遍历 Web,力求在限定的爬行周期内收集到尽可能多的 Web 页面。p2p 节点聚类中,我们建立兴趣爬虫代理。爬虫代理主要任务不是下载节点发布信息的内容,而是通过计算节点间的信息相似度为节点建立节点簇提供数据。

向量空间模型(VSM)^[16]广泛应用于文本分类和聚类,并通过文本的特征出现的频率(词频)向量表示文本。如文本 $d_i = \langle tf_{1i}, tf_{2i}, \dots, tf_{ki}, \dots, tf_{mi} \rangle$ 。其中, tf_{ki} 表示特征词 k 在文本 d_i 中出现的次数, n 表示文本中的特征词数。

特征权重基于 tf/idf 计算。如第 k 个特征的权重

$$w_{ki} = \frac{\log(tf_{ki} + 1) \times \log\left(\frac{n+1}{n_k}\right)}{\sqrt{\sum_{i=1}^m [\log(tf_{ki} + 1) \times \log\left(\frac{n+1}{n_k}\right)]}}$$

其中 n 为文本总数, n_k 为出现特征词 k 的文本数。此时, 文本 i 的特征权重向量表示为

$$w_i = \langle w_{1i}, w_{2i}, \dots, w_{ki}, \dots, w_{mi} \rangle$$

我们利用爬虫代理计算文本间的相似度 $s(i, j)$ (表示第 i 个文本和第 j 个文本间的相似度)。使用余弦(Consine)相似度计算公式, 如式

$$s(i, j) = \frac{\sum_{k=1}^m w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^m (w_{ki})^2 \cdot \sum_{k=1}^m (w_{kj})^2}}$$

$s(i, j)$ 的取值范围为 $[0, 1]$ 。如果该值越大, 表示文本越相似。我们将此作为被爬虫代理访问节点得分的第一部分。另外, p2p 中, 每个节点都与多个节点相连接, 称这样的节点为它的直接邻居。通过相似度计算公式, 可以计算邻居节点间文本的相似度, 由这些相似度构成节点得分的第二部分。

根据爬虫代理所得的文本相似度, 我们如下计算节点的得分:

$$R_i(j) = (1 - \gamma)s(i, j) + \gamma \sum_{k=1}^N s(j, k) / N \quad (1)$$

式(1)中, $R_i(j)$ 为相对于节点 i 某个主题兴趣节点 j 的得分。 $s(j, k)$ 为节点 j 与其邻居节点 k 之间的文本相似度, N 为节点 j 的邻居中文本相似度大于某个阈值的节点数。 $\gamma \in [0, 1]$, 为折扣因子, 用于调节两部分的权重。

爬虫代理每次爬行节点后, 节点的得分情况就要发生变化。刚开始时, 节点还没有建立其节点聚类, 因此其与邻居节点间的相似度还没有对其得分产生贡献。随着节点聚类的增加, 上式中的第二部分逐步产生数据, 改变节点的得分, 因此称其为基于增量的学习。

4.2 信息查询机制

4	2	5	1	3	6	7
0.85	0.78	0.71	0.68	0.67	0.58	0.56

图1 节点得分表

根据式(1), 爬虫代理的属主节点可以得到相关节点基于相似度和增量学习的得分。按照得分, 属主节点将其它节点按照得分降序方式排列, 同时直接忽略得分小于设定阈值的节点信息。这个排序并不是在爬虫代理爬完所有相关节点后才建立, 而是爬虫代理每爬过一个节点, 就将结果信息发送到属主节点。属主节点根据阈值决定取舍, 同时将保留的信息

按降序插入到节点得分信息表相应位置。得分信息表根据爬虫代理发回的信息不断变化。

图1中表示节点簇中的7个节点得分的降序表。如果某个查询信息检索该节点, 节点首先计算本节点文本与查询信息的相似度情况, 若找到合适的答案, 查询就在本节点簇内进行。否则根据系统资源定位机制, 通过结构化层, 将消息发送到相应的 SP 节点, SP 节点根据节点簇提供的簇内信息, 向对应的节点簇的主节点转发消息。接到消息的主节点, 首先判断本身能否提供合格答案。若能提供满足语义的信息, 查询按照该节点所建立的节点簇信息表, 同时按得分情况依次向簇内节点转发消息, 计算是否找到合格答案。如果消息连续3次转发不能找到合格答案, 说明该转发路径失败, 结束沿该路径的消息转发。

5 评价标准及方法

文本分类中, 查全率和查准率是两个重要的指标。p2p 信息检索中, 我们要考察查询返回结果的命中率(查全率), 同时要考察返回结果的效率以及网络资源的利用率。

查询受系统速度的影响。相同时间内, 算法运行于速度快的机器将返回更多结果。为此, 实验必须在相同环境下, 比较相同时间段内返回结果的数量。我们给出查全率 R_c 、查准率 P_c 和 F-measure 的计算公式:

$$R_c = \frac{Nr_{[0, T]}}{No_{[0, T]}}, P_c = \frac{Nr_{[0, T]}}{N}, F\text{-measure} = \frac{2R_c P_c}{R_c + P_c}$$

其中, $Nr_{[0, T]}$ 表示从起始时间 0 经过 T 时间检索到的合格答案数, $No_{[0, T]}$ 为该时间段内在线的合格答案数, N 为返回的结果总数。

我们设计查询在节点簇内、簇间及 Gnutella 进行。簇内查询发生在节点查询信息与节点发布信息具有高相似度的情况。由于目标节点就在本节点维护的节点簇内, 查询比较方便。传统意义上的 p2p 基于主题的查询一般属于此类查询, 称为 RIC(Retrieval In Cluster)。簇间查询时, 查询节点发布的信息与查询信息不属于一类, 这种情况发生在用户对另外类的信息产生兴趣。查询需要在另外节点簇内进行, 称为 RAC(Retrieval Among Cluster)。另外, 我们比较在 Gnutella 上进行信息查询的情况。由于 Gnutella 代表了非结构化 p2p 网络, 并且现在很多流行的与 Gnutella 相似的资源发布与查询路由策略, 称为 RIG(Retrieval In Gnutella)。我们在 RIC、RAC 和 RIG 中执行相同的查询, 并限定相同的查询时间间隔。

6 实验分析

实验平台基于 JAVA 开发, 安装在运行 Windows 2000 的机器上, 仿真 600 个节点。我们在每一个节点上发布了至少两个主题兴趣的文档, 整个系统中的文档兴趣主题共有 96 个, 每个主题文件数从 31 到 294 不等, 文件总数 14560。每个文件用 100 个关键词索引。基本数据如表 1 所示。

表1 实验基本数据

节点数	SP 数	主题数	关键词	文件数	主题最少文件数	主题最少文件数	节点平均文件数	δ	阈值
600	96	96	100	14560	31	294	29	0.15	0.45

我们依据表 1 的数据考察 RIC、RAC 和 RIG 中的查全率、网络传输效率。经过 20 次的平均后, 我们得到表 2 所示

的结果。

表2 返回结果数据的时间比较

返回答案数量\时间(ms)\类型	RIC	RAC	RIG
1	43	47	53
2	45	53	65
4	48	56	74
6	52	62	90
8	56	64	96
10	58	69	107
12	64	70	122
14	69	74	158
16	71	79	201

表2表明,返回结果相同时,RIC所用的时间最少,这是因为查询在本节点簇内进行。同时RAC所用时间大于RIC所用时间,因为查询首先需要通过SP节点查询到相应的节点簇,并由SP将节点簇的信息发送到查询消息提交节点,该节点再将查询消息发送到相应的节点簇。一旦查询提交节点与相应的节点簇建立了联系,以后的查询就变成了簇内查询,所以我们看出RIC和RAC的查询时间差没有大的变化。RIG的查询模式不同于前两种,由于通过消息的泛洪实现查询,效率比较低,当需要更多返回结果时,所需时间变得越来越长。

网络搜索中,用户无法估计查询的效果,此时需要用查全率。我们不是通过设置一个固定的查询时间,而是通过设置查询消息的TTL值,考察查询所得结果占发布该类文件的比例。平均20次实验,得表3所示结果。

表3 不同情况下的测量指标($\gamma=0.15$)

	R_c	P_c	F-measure
RIC	32.09%	77.90%	0.4546
RAC	29.82%	78.05%	0.4315
RIG	4.97%	77.84%	0.0934

表3表明,三种情况的查准率大致相同,这是因为都采用了相同的文本内容相似度计算方法。查全率方面,RIG要明显差于RIC和RAC,RIC的结果最好。这是由于相同查询时间内,RIC和RAC通过聚类,将相似的节点归到一个节点簇内,大大提高了查询效率。而RIG采用泛洪方式,盲目在网络中转发查询消息,命中答案的几率低,从而返回的结果就少。从F-measure可以看出三者的区别。 γ 的设置影响查询结果,实验表明 $\gamma=0.15$ 比较好。

结论 本文研究了p2p网络中节点聚类情况下的信息查询。当前大多数p2p文件共享都是基于主题词和文件名匹配的信息查询,缺乏文本语义方面的考虑。随着网络规模的扩大,这种方式并不能充分发挥p2p网络的优势。我们知道,文本分类和聚类技术已经取得了很好的成功,如果能够借助于相关的技术,将发布在节点上的信息按照一定的方式建立节

点聚类,信息检索只需在类间进行,将会大大提高查询效率。基于这样的思想,我们采用向量空间模型(VSM)描述发布信息,并通过结构化连接的SPs管理基于不同主题的信息节点。p2p节点空闲时,向网络中具有相同主题的节点发送兴趣爬虫代理。爬虫代理根据增量学习算法计算所爬节点得分,并把满足设定阈值节点的相关元数据发送到爬虫主节点,据此建立节点聚类。

簇内各节点按照得分降序排列,接到查询信息后,首先从本节点查询,同时依次向簇内节点发送查询消息。由于p2p网络的动态性,节点随时都可能加入和退出,从而增加了网络管理的复杂性。节点簇建立起来后,簇内节点可能不在线。此时,向该类节点发送查询消息已经没有意义。可通过建立活动窗口,屏蔽掉这类节点,减少网络中的查询消息数量,提高网络利用率和查询效率。目前这方面的工作比较少,我们试着将Web爬虫技术应用到p2p信息查询中,建立活动窗口,提高查询效率。

参考文献

- 1 Napster home page. <http://www.napster.com>
- 2 Gnutella home page. <http://gnutella.wego.com>
- 3 Yang B, Garcia-Molina H. Efficient search in peer-to-peer networks. In: proc. of intl. conf., ICDCS, 2002
- 4 Triantallou P, Xiruhaki C, Koubarakis M, et al. Towards high performance peer-to-peer content and resource sharing systems. In: proc. of the intl. conf. on innovative data systems research (CDIR), 2003
- 5 Rowstron A, Druschel P. Storage management and caching in PAST: A large-scale persistent peer-to-peer storage utility. In: ACM SOS'01, Banff, Canada, Oct. 2001
- 6 Can project home page. <http://www.icir.org/sylvia/>
- 7 Chord project home page. <http://www.pdos.lcs.mit.edu/chord/>
- 8 Lu J, Callan J. Content-based retrieval in hybrid peer-to-peer networks. CIKM'03, Nov. 2003
- 9 <http://www.nms.lcs.mit.edu/HotNets-II/papers/structella.pdf>
- 10 Risson J, Moors T. Survey of research towards robust peer-to-peer networks: search methods. [Technical Report UNSW-EE-P2P-1-1]. University of New South Wales, Sydney, Australia. Sept. 2004
- 11 Milojicic D S, Kalogeraki V, Lukose R, et al. Peer-to-peer computing 2002 HP. <http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf>
- 12 Jxta project homepage. <http://www.jxta.org/>
- 13 Crespo A, Garcia-Molina H. Routing indices for peer-to-peer systems. icdc's'02, July. 2002
- 14 Lu Z, McKinley K S. The effect of collection organization and query locality on information retrieval system performance and design. Book chapter in advances in information retrieval, Kluwer, New York, 2000
- 15 Menczer F, Pant G, Srinivasan P. Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology, Forthcoming, online at <http://www.informatics.indiana.edu/fil/Papers/TOIT.pdf>, 2003
- 16 Salton G. Automatic Information Organization and Retrieval. New York: McGraw-Hill, 1968

(上接第183页)

分来分析概念,通过对概念的层级递阶来进行概念的泛化与细化,通过格结构使概念在递阶方面忽略不必要的冗余,达到更高的效率。

参考文献

- 1 Yao Y Y, Chen Y H. Rough Set Approximations in Formal Concept Analysis. NAFIPS, 2004
- 2 Yao Y Y. A Partition Model of Granular Computing. LNCS Transactions on Rough Sets
- 3 Pawlak. Rough Sets. Int J Computer and Information Sci, 1982,

11:341~356

- 4 Ganter B, Wille R. Conceptual scaling. In: Roberts F, ed.: Applications of combinatorics and graph theory to the biological and social sciences, New York: Springer-verlag, 1989. 139~167
- 5 Prediger S. Logical scaling in formal concept analysis. In: Lukose D, eds. Conceptual Structures: fulfilling Peirce's Dream. Proc. of the ICCS'97, LNAI 1257, Springer, Berlin, 1997. 332~341
- 6 胡可云, 陆玉昌, 石纯一. 粗糙集理论及其应用进展[J]. 清华大学学报(自然科学版), 2001, 41(1): 64~68
- 7 王俊红, 梁吉业. 概念格与粗糙集. 山西大学学报(自然科学版), 2003, 26(4): 307~310