第35卷 第10期 计 算 机 工 程 2009年5月
Vol.35 No.10 Computer Engineering May 2009
·博士论文· 文章编号：1000—3428(2009)10—0014—04 文献标识码：A 中图分类号：TP391

对称和非对称词语聚类模型的比较研究

孙越恒，曹桂宏，侯越先
(天津大学计算机科学与技术学院，天津 300072)

摘 要：词语聚类是语音识别、智能信息检索等领域的一个重要的自然语言处理问题。实现基于互信息的对称聚类模型，并针对该模型未考虑词语顺序的缺陷，提出一种新的非对称聚类模型。按照聚类词相对其他词语的位置关系，该模型分为2个子模型，即条件聚类模型和预测聚类模型。在大规模数据集上的实验表明，相对于对称聚类模型，非对称聚类模型是一种更为有效的词语聚类模型。

关键词：词语聚类；对称聚类模型；非对称聚类模型

Comparative Research on Symmetric and Asymmetric Word Clustering Models

SUN Yue-heng, CAO Gui-hong, HOU Yue-xian
(School of Computer Science and Technology, Tianjin University, Tianjin 300072)

【Abstract】 Word clustering is one of important natural language processing issues in speech recognition and intelligent information retrieval, etc. This paper presents a symmetric clustering model based on mutual information. For the model not taking the order of words into account, it proposes a new asymmetric clustering model including two sub models, conditional clustering model and predictive clustering model. Experimental results on large scale data set show that compared with the symmetric clustering model, the asymmetric clustering model is a more effective one for clustering words.

【Key words】 word clustering; symmetric clustering model; asymmetric clustering model

1 概述

词语聚类，即将一个给定的词 w_i 分到某个词类 c_i 中去。在自然语言处理的研究中，词语聚类算法是被广泛研究的课题。由一些语义或语法相近的词组成的词类可以看成是纷繁的个别词语现象到语言的一般概念的映射，而这些概念是更加本质的，更具有一般性和稳定性，这对于语音识别、智能检索等许多自然语言处理领域均有实用价值。

词语聚类按方法主要分为2类：(1)基于知识的聚类，即基于规则的聚类；(2)数据驱动，即基于统计的聚类。在基于知识的聚类方法中，根据词的语法或者语义信息对词进行聚类[1-3]。一个很典型的例子就是按词性分类。但是由于在语言学界对词类体系没有一个共识，或者说没有一个统一的标准，因此按词性对词分类没有取得令人满意的结果。早期研究表明，这种方法会使得模型的复杂度很高[4]。然而如果有某个特定领域的知识，那么将语法功能相似的词语聚在一起能够取得较好的效果[3]。

所谓数据驱动的聚类，就是不用任何语法和语义知识，完全利用语料的统计信息对词进行自动聚类[5-6]。该方法通常利用EM算法，不断降低聚类之后整个文本的复杂度，最后找到一种最优的聚类结果。最优结果的搜索策略通常是贪心算法，因而是局部最优而非全局最优。这种方法大大降低了计算复杂性，从而使得对词语自动聚类成为可能。

2 基于互信息的对称词语聚类

假设将词典 D 里的词分到 C 个类里面去。首先定义一个多对一的映射函数 π，它能够将词语 w_i 映射到对应的类 c_i 上，即 $\pi(w_i)=c_i$。在此基础上可以定义一个基于类的n-gram模型。对所有的 k，都有：

$$p_r(w_k \mid w_1 w_2 \cdots w_{k-1}) = p(w_k \mid c_k)p(c_k \mid c_1 c_2 \cdots c_{k-1}) \quad 1 \le k \le n$$

显然，对于每一个这种形式的n-gram模型都会有 $c^n-1+V-C$ 个独立参数，其中 V 为所有的词的个数。其实这是显然的，因为在上述模型中，$p(c_k \mid c_1 c_2 \cdots c_{k-1})$ 有 c^n-1 个独立参数，而 $p(w_k \mid c_k)$ 又有 V-C 个独立参数。与纯粹的n-gram模型相比，这种形式的聚类语言模型含有较少的参数。

假定已经有了一个长度为 T 的训练语料 t_1^T。下面讨论基于类的bigram模型，如下式所示：

$$p(w_k \mid w_{k-1}) = p(w_k \mid c_k)p(c_k \mid c_{k-1}) \tag{1}$$

用最大似然估计法计算模型的参数：

$$p(c_k \mid c_{k-1}) = \frac{C(c_{k-1},c_k)}{\sum_c C(c_{k-1},c)}, \quad p(w_k \mid c_k) = \frac{C(w_k)}{C(c_k)}$$

下面计算映射函数 π 的最大似然函数：

$$\begin{aligned} L(\pi) &= \sum_{(w_{k-1},w_k)} \frac{C(w_{k-1},w_k)}{T-1} \mathrm{lb} p(w_k \mid w_{k-1}) = \\ &\sum_{(w_{k-1},w_k)} \frac{C(w_{k-1},w_k)}{T-1} \mathrm{lb} p(c_k \mid c_{k-1})p(w_k \mid c_k) = \\ &\sum_{(c_{k-1},c_k)} \frac{C(c_{k-1},c_k)}{T-1} \mathrm{lb} \frac{p(c_k \mid c_{k-1})}{p(c_k)} + \\ &\sum_{w_k} \frac{\sum_w C(ww_k)}{T-1} \mathrm{lb} p(w_k \mid c_k)p(c_k) \end{aligned} \tag{2}$$

基金项目：国家自然科学基金资助项目(60603027)
作者简介：孙越恒(1974－)，男，讲师、博士，主研方向：自然语言处理，信息检索与挖掘；曹桂宏，硕士；侯越先，副教授、博士
收稿日期：2009-02-23 **E-mail：**yhs@tju.edu.cn

因为 T 足够大，故 T-1 近似等于 T，所以有 $\frac{C(c_{k-1},c_k)}{T-1}=p(c_{k-1},c_k)$，且 $\frac{\sum_w C(ww_k)}{T-1}=p(w_k)$，于是：

$$L(\pi)=\sum_{w_k} p(w_k)\mathrm{lb}p(w_k)+\sum_{(c_{k-1},c_k)} p(c_{k-1},c_k)\mathrm{lb}\frac{p(c_{k-1},c_k)}{p(c_{k-1})p(c_k)}=$$
$$-H(w)+\sum_{(c_{k-1},c_{k-2})} MI(c_{k-1},c_k) \quad (3)$$

其中，$H(w)$是 unigram 模型的熵，而式子的另一部分则是相邻的类的互信息之和。由于 $H(w)$和分类的结果无关，因此最大化映射函数 π 就等价于最大化相邻类的互信息之和。但是众所周知，没有一种算法能够在可控制的计算复杂度内计算出这个映射函数，所以使用了一种局部取优的贪心算法。该算法能够有效地降低计算复杂度，而且性能也不致受到很大的影响。

算法过程如下：

(1)将每个词分配给不同的类，此时就有$|V|$个类，$|V|$为不同词的个数。

(2)计算相邻的类之间的互信息的和。

(3)合并邻近的类，计算合并之后的相邻的类的互信息之和。选出这样的 2 个类，使得合并它们之后，互信息之和的损失最少。

(4)判断剩下的类数目是否合乎要求，如合乎要求，则输出聚类结果，退出程序。否则，跳转到(3)继续执行。

在上面的算法中，并没有在每次合并之后调整每个类里面的元素，而是认为类里面的元素已经达到最优了，所以说该算法实际上是局部最优的。因此，算法的过程就是每次合并 2 个类，即减少 1 个类。所以要想得到 C 个类的话，应该重复运行$|V|$-C 次才能达到要求。

3 非对称词语聚类

下面从 2 个角度来考虑聚类，首先按照一个词前面相邻的词来聚类，例如词对<w_1, w_2>，按照 w_1 来对 w_2 聚类，称之为预测聚类模型；然后按照该词后面相邻的词来对该词聚类，例如词对<w_1, w_2>，按照 w_2 对 w_1 聚类，称之为条件聚类模型。

3.1 条件聚类模型

在条件聚类的过程中，要最小化聚类结果的熵。这很容易理解，因为从信息论的角度来说，把词聚成几个类，每个类里面的词都是无区别地对待，所以必然会损失一些信息，这就会使得这个模型的熵增加，而增加最少的就是信息损失最少的，正是所求的。

最小化熵就等价于最小化：

$$H(\pi)=-\sum_v c(Wv)\mathrm{lb}p(v|W) \quad (4)$$

其中，v 是当前的词；而 W 则是与它相邻的词所属于的类。

在第 2 节对称聚类模型的算法中所使用的是自下而上的聚类方法。从以上分析可以看到这种方法的计算复杂度相当大。为了降低复杂度，在这里使用了一种自上而下的方法。首先让所有的词都属于 1 个类，然后将这个类分成 2 类，分类的方法是先随机地将这些词分成 2 类，然后再将这 2 个类的词换进换出，以使得结果的熵减小，直到最后收敛为止，这样就将 1 个类分成了 2 个类。如果类的数目太少了，可以将分出来的类再重复上面的操作，直到分类结果满足要求为止。这样分类的结果实际上就是一个平衡二叉树。每一层都是一个分类结果，类的数目就是 2 的某次幂。

事实上，要最小化 $H(\pi)=-\sum_v c(Wv)\mathrm{lb}p(v|W)$，即最大化 $-H(\pi)=\sum_v c(Wv)\mathrm{lb}(v|W)$。假设将一个词 x 加到了类 W 中，那么新的熵怎么计算呢？假设新的类 $W+x$(在原来的类的基础上再增加 x)为 X。则有：

$$\sum_v c(Xv)\mathrm{lb}p(v|X)=\sum_{v|c(x,v)>0} c(Xv)\mathrm{lb}p(v|X)+$$
$$\sum_{v|c(x,v)=0} c(Xv)\mathrm{lb}p(v|X) \quad (5)$$

在式(5)中，左半部分很好计算，因为这样的 v 是有限的，而且很容易找到。下面只讨论右半部分值的计算。

$$\sum_{v|c(x,v)=0} c(Xv)\mathrm{lb}p(v|X)=\sum_{v|c(x,v)=0} c(Wv)\mathrm{lb}p(v|W)\frac{c(W)}{c(X)}=$$
$$\sum_{v|c(x,v)=0} c(Wv)\mathrm{lb}p(v|W)+$$
$$\frac{c(W)}{c(X)}\sum_{v|c(x,v)=0} c(Wv) \quad (6)$$

注意到：

$$\sum_{v|c(x,v)=0} c(Wv)\mathrm{lb}p(v|W)=\sum_v c(W|v)\mathrm{lb}p(v|W)-$$
$$\sum_{v|c(x,v)>0} c(W|v)\mathrm{lb}p(v|W) \quad (7)$$

同时又有：

$$\sum_{v|c(x,v)=0} c(Wv)=\sum_v c(Wv)-\sum_{v|c(x,v)>0} c(Wv)=$$
$$c(W)-\sum_{v|c(x,v)>0} c(Wv) \quad (8)$$

将式(7)和式(8)代入式(6)，得到：

$$\sum_{v|c(x,v)=0} c(Xv)\mathrm{lb}p(v|X)=\sum_v c(Wv)\mathrm{lb}p(v|W)-$$
$$\sum_{v|c(x,v)>0} c(Wv)\mathrm{lb}p(v|W)+$$
$$(\mathrm{lb}\frac{c(W)}{c(X)})(c(W)-\sum_{v|c(x,v)>0} c(Wv)) \quad (9)$$

其中，$\sum_v c(Wv)\mathrm{lb}p(v|W)$ 实际上就是原来的负熵，所以这部分不用计算。通过上面的式子就能计算换进换出一个词之后的熵，这样就可以通过最小化熵来找到合适的类。

3.2 预测聚类模型

预测聚类相当于给定一个词来预测下一个词属于哪一个类。下面证明可以使用条件聚类模型的方法来实现预测聚类。设语料的总长度为 N，那么条件聚类模型就相当于要最大化 $\prod_{i=1}^{N} p(w_i|W_{i-1})$，其中 w_i 是第 i 个词，而 W_{i-1} 是第 i-1 个词所在的类。假设在预测聚类模型中 w_i 属于 W_i 类，那么就相当于要最大化：

$$\prod_{i=1}^{N} p(W_i|w_{i-1})p(w_i|W_i)=\prod_{i=1}^{N}\frac{p(w_{i-1}W_i)p(W_iw_i)}{p(w_{i-1})p(W_i)}=$$
$$\prod_{i=1}^{N}\frac{p(w_i)}{p(w_{i-1})}p(w_{i-1}|W_i)$$

因为 $\frac{p(w_i)}{p(w_{i-1})}$ 与聚类的结果无关，所以要最大化 $\prod_{i=1}^{N} p(W_i|w_{i-1})p(w_i|W_i)$，实际上只需要最大化 $\prod_{i=1}^{N} p(w_{i-1}|W_i)$，这就相当于把条件聚类模型颠倒过来的结果。所以用于条件聚类模型的工具在预测聚类模型中都可以使用。

将条件聚类模型和预测聚类模型结合起来，非对称的词聚类算法包括如下步骤：

(1)利用条件聚类模型，对目标词语集合进行聚类，得到聚类结果 A。

(2)利用预测聚类模型，对目标词语集合进行聚类，得到聚类结果 B。

(3)融合 2 种聚类结果。

具体方法如下：

(1)判断 2 种聚类结果 A 和 B 中，哪些类是同一类，其标准是具有最多相同词语个数的 2 个类为相同类，例如，若 A 中的类 a_1 和 B 中的类 b_2 具有最多相同词语个数，即认为这 2 个类是同一类。

(2)将相同的词语提取出来，形成一个新类 c_1。

(3)对于 2 个类中不同的词，比较这些词属于 c_1 类的概率是否大于属于其他类的概率，若是，则保留在 c_1 类中；否则，保留在其他类中。

(4)依次处理所有对应类，得到最终结果。

4 实验结果与分析

4.1 实验数据

本文采用了两部分的数据：

(1)3 年的 Wall Street Journal 语料，共 260 741 个词；

(2)含有 50 180 个词的中文语料。

2 种聚类模型在聚类后得到的类数均是 64。

4.2 评价指标

一般文本聚类采用的评价指标是熵或 *F-measure* 值，其中需要提前人工制定作为评分标准的答案。但对于本实验而言，由于词语数量众多，很难人为确定真实类别，即便是由专家来定，每人也可能有不同的分类标准。因此本实验采用了一种折中方案：从 2 种聚类模型的结果中，人工选取最为相近的“类对”(每个类对中的 2 个类看成同 1 个类)，确定类对应该拥有的词语，作为该类对的标准词语集合，假设其个数为 N。考查类对中的一个类，其中拥有标准词语的个数为 tp，非标准词语的个数为 fp，则对于该类而言，其精确率(P)、召回率(R)和 *F-measure* 值分别为

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{N}$$

$$F - measure = \frac{2PR}{P + R}$$

对于其他的类采取同样的计算方式，可分别求出上述数据。

4.3 结果分析

表 1 和表 2 分别是对称聚类模型在 Wall Street Journal 语料上的测试结果(记为 SC_{WST}，其中 *SC* 表示对称聚类，*WST* 表示 Wall Street Journal 语料)和非对称聚类模型在中文语料上的测试结果(ASC_{CP}，其中 *ASC* 表示非对称聚类，*CP* 表示中文语料)。另外 2 种聚类结果，即 SC_{CP} 和 ASC_{WST} 未列出。表 3 和表 4 分别表示 2 种聚类模型在不同语料上的 *F-measure* 值的比较。F_{SC} 表示对称聚类得到的 *F-measure* 值，F_{ASC} 表示非对称聚类得到的 *F-measure* 值。

表 1　对称聚类模型在 Wall Street Journal 语料上的测试结果 SC_{WST}

类	词语
A	Friday Monday Thursday Wednesday Tuesday Saturday weekends Sundays Saturdays
B	June March July April December October November September August
C	Down backwards ashore sideways southward northward overboard aloft downwards
D	Water gas coal liquid acid sand carbon steam shale iron
E	Great big sudden mere sheer gigantic lifelong scant colossal
F	Man woman boy girl lawyer doctor guy farmer teacher citizen
G	American Indian European Japanese German African Catholic
H	Pressure temperature permeability density porosity stress
I	Machine device controller processor CPU printer spindle subsystem compiler plotter

表 2　非对称聚类模型在中文语料上的测试结果 ASC_{CP}

类	词语
A	走，飞跑，奔腾，高攀，推翻，跳动，流淌，吝惜，凄然，…
B	老师，先生，小姐，同志，父亲，母亲，讨伐，发誓，…
C	篮球，棒球，乒乓球，铅球，地球，宴会，…
D	进行，建立，提出，实现，取得，提供，出现，得到，形成，发生，发挥，产生，完成，获得，发表，创造，召开，出席，所有，…
E	继续，再次，重新，坚决，第一次，多次，经常，纷纷，突然，立即，刚刚，逐渐，尽快，主动，从中，亲自，彻底，提前，反复，马上，…
F	汽车，石油，建筑，制造，加工，食品，化学，化工，机械，本地，广告，航空，制作，航天，示范，电力，服装，纺织，钢铁，走私，…
G	重要，主要，群众，一定，基本，重大，实际，一切，高度，人类，一般，具体，根本，自然，核心，特殊，自身，客观，各自，唯一，最好，自我，周围，军人，绝对，历史性，彼此，最低，…
H	广州，贫困，深圳，天津，纽约，南京，厦门，重庆，巴黎，东北，西安，福州，长江，华盛顿，东京，成都，大连，珠海，武汉，沿海，西南，南方，黄河，整顿，山区，…
I	所谓，称为，誉为，可谓，叫做，称之为，致函，评为，人称，发给，称作，素有，号称，鼓吹，当作，…
J	过去，以后，之后，当时，一天，后来，如今，为此，另外，当年，晚上，不久，面前，之前，身上，这时，拒绝，中间，随后，那天，…
K	积极，不断，充分，认真，广泛，深入，正确，有效，真正，逐步，健康，明显，迅速，严格，明确，顺利，普遍，热烈，热情，合理，及时，切实，更好，有力，大大，显著，自觉，相应，…

表 3　2 种聚类模型在 Wall Street Journal 语料上 *F-measure* 值比较

聚类模型	P_1	P_2	P_3	P_4	P_5	P_6
F_{SC}	0.825	0.679	0.716	0.679	0.845	0.789
F_{ASC}	0.839	0.733	0.752	0.754	0.895	0.855
聚类模型	P_7	P_8	P_9	P_{10}	P_{11}	P_{avg}
F_{SC}	0.784	0.765	0.713	0.746	0.831	0.761
F_{ASC}	0.862	0.814	0.758	0.812	0.906	0.816

表 4　2 种聚类模型在中文语料上 *F-measure* 值比较

聚类模型	P_1	P_2	P_3	P_4	P_5	P_6	P_7
F_{SC}	0.462	0.500	0.524	0.526	0.418	0.602	0.563
F_{ASC}	0.530	0.612	0.608	0.598	0.465	0.750	0.582
聚类模型	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{avg}	
F_{SC}	0.535	0.478	0.496	0.465	0.625	0.516	
F_{ASC}	0.660	0.534	0.574	0.543	0.716	0.589	

表 3 与表 4 反映出非对称聚类模型在 5 种语料上的测试结果均好于对称模型。在对称聚类中，目标函数是相邻的类之间的互信息之和，而互信息具有对称性，即 $MI(w_i, w_j) = MI(w_j, w_i)$，因而这种聚类算法没有考虑到词与词的相对顺序，而仅仅考虑到了词与词是否相邻。举个例子，如果语料中其他的信息相同，那么下面 2 句话的聚类结果相同：

*S*1：小狗很小

*S*2：小很狗小

因为在上面的 2 句话中，仅仅只是调换了“狗”和“很”的顺序，其他的信息都没有改变，而在计算目标函数的时候，这 2 个句子对于目标函数的贡献都是一样的，因为它们含有相同的词对：

*S*1 的词对为：　小狗　　狗很　　很小

*S*2 的词对为：　狗小　　很狗　　小很

当然，在真实语料中出现第 2 句话的概率很小，但是诸如此类的情形还是可能出现的。重点是，词与词之间的相对顺序的确会影响到聚类的结果。例如，如果按照词语后面紧跟的词来对词语聚类，那么“is”和“are”不可能聚为一类，因为“is”后面跟的一般是定冠词或者是不定冠词，但是“are”后面一般直接跟名词。“apple”和“banana”会聚为一类，因为“apple”和“banana”后面紧跟的单词可能一样。但是如果根据词前面的相邻的词来对词聚类，那么“is”和“are”可以聚为一类，因为它们前面可能都接一个名词型主语；但是“apple”和“banana”却不能聚为一类，因为“apple”前

面接“an”，而“banana”前面接“a”。非对称聚类模型综合考虑了上述2种不同情形，因而可得出正确的聚类结果。

注意到，事实上在上面的聚类过程中，并没有用到任何语言学方面，包括语法和语义方面的知识。但是有趣的是，聚类的结果显示，在这样聚出来的类中，很多聚在相同类中的元素都在语料中扮演相同或者相似的语法角色，或者具有相近意义。这是因为，从本质上来说，数据驱动的方法，抑或基于统计的方法都是通过计算词的共现信息来聚类的，部分语法或语义相似的词在训练语料中的分布具有很大的相似性，因而很容易聚在相同类中。例如，在表2中，A类大部分都是动词，B类和C类均是名词。更细一点看，A类中大部分的词都是表示一种动作，B类中的词一般都是表示头衔，C类中的词表示一种球类运动。同时也可发现，如果单从语义的角度考虑的话，有些词则似乎是聚错了，例如A类中的“凄然”是形容词而不是动词，B类中的“讨伐”和“发誓”是动词而非名词，C类中的“地球”和“宴会”是名词，但是它们并不是表示一种球类运动。所以，虽然聚类结果中的大部分在语义上都是有意义的，但是有些词的出现却出人意料。其实这可以从数据稀疏的角度来解释，如果某个词偶尔有非正常的分布，例如动词出现在本该名词出现的地方，由于数据较少，这个非正常的分布将严重影响该词的聚类取向，使之聚到错误的类中。

5 结束语

对称聚类模型通过最大化相邻词类之间的互信息来实现词语聚类，这种聚类算法没有考虑到词语的相对顺序。本文将词语顺序考虑在内，分别提出了条件聚类模型和预测聚类模型，并将两者融合到一起，形成一种非对称词语聚类模型。在2个大规模数据集上的测试表明，相对于对称聚类模型而言，非对称聚类模型是一种更为有效的词语聚类模型。

本文当前的研究工作默认一个词只属于一个类，但是由于自然语言的复杂性，多义词现象非常普遍，也就是说，一个词可能会聚到不同的类中，即所谓的软聚类。从理论上来说，软聚类更加符合语言实际，因此，探讨硬聚类和软聚类的理论关系以及适应软聚类的模型、算法将是今后的研究重点。

参考文献

[1] Heeman P. POS Tags and Decision Trees for Language Modeling[C]//Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Maryland, USA: [s. n.], 1999: 129-137.

[2] Heeman P, Allen J. Incorporating POS Tagging into Language Modeling[C]//Proc. of the 5th European Conference on Speech Communication and Technology. Rhodes, Greece: [s. n.], 1997: 2767-2770.

[3] Issar S, Ward W. Flexible Parsing: CMU's Approach to Spoken Language Understanding[C]//Proc. of the ARPA Spoken Language Technology Workshop. Princeton, New Jersey, USA: [s. n.], 1994: 53-58.

[4] Niesler T R, Whittaker E W D, Woodland P C. Comparison of Part-of-speech and Automatically Derived Category-based Language Models for Speech Recognition[C]//Proc. of International Conference on Acoustics, Speech, and Signal Processing. Seattle, WA, Australia: [s. n.], 1998: 177-180.

[5] Brown P, Pietra V D, Souza P D, et al. Class-based N-gram Models of Natural Language[J]. Computational Linguistics, 1992, 18(4): 467-479.

[6] Wang Bo, Wang Houfeng. A Comparative Study on Chinese Word Clustering[C]//Proc. of ICCPOL'06. Berlin, Germany: [s. n.], 2006: 157-164.

编辑　任吉慧

(上接第13页)

参考文献

[1] 任丰原, 黄海宁, 林　闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1291.

[2] 杨少军, 史浩山, 黄　睿. 无线传感器网络移动 Agent 路由算法的研究与仿真[J]. 系统仿真学报, 2007, 19(2): 388-395.

[3] 周四望, 林亚平, 聂雅琳, 等. 无线传感器网络中基于数据融合的移动代理曲线动态路由算法研究[J]. 计算机学报, 2007, 30(6): 894-904.

[4] Akyildiz F, Cayirci E, Sankarasubramaniam Y, et al. Wireless Sensor Networks[J]. A Survey Computer Networks, 2002, 38(4): 393-422.

[5] Kumar C Y C, Kumar S. Sensor Networks: Evolution Opportunities and Challenge[J]. Proceedings of the IEEE, 2003, 91(8): 1247-1256.

[6] Wook C, Das S K. A Novel Famework for Energy Conserving Data Gathering in Wireless Sensor Networks[C]//Proc. of the 24th Conference on Computer Communications. Miami, USA: IEEE Press, 2005: 1985-1996.

[7] Qi Hairong, Iyengar S S, Chakrabarty K. Multiresolution Data Integration Using Mobile Agents in Distributed Sensor Networks[J]. IEEE Transactions on Systems, Man and Cybernetics(Part C): Applications and Reviews, 2001, 31(3): 383-291.

[8] Qi Hairong, Xu Yingyue, Wang Xiaoling. Mobile Agent Based Collaborative Signal and Information Processing in Sensor Networks[J]. Proceedings of the IEEE, 2003, 91(8):1172-1183.

[9] Avramopoulos I C, Anagnostou M E. Optimal Component Configuration and Component Routing[J]. IEEE Trans. on Mobile Computing, 2002, 1(4): 303-312.

[10] Migas N, Buchanan W J, McAartney K A. Mobile Agents for Routing, Topology Discovery and Automatic Network Reconfiguration in Ad-Hoc Networks[C]//Proc. of ECBS'03. Huntsville, AL, USA: IEEE Press, 2003: 200-206.

[11] Lu Shiyong, Xu Chengzhong. A Formal Framework for Agent Itinerary Specification, Security Reasoning and Logic Analysis[C]//Proc. of ICDCSW'05. Columbus, Ohio, USA: [s. n.], 2005: 580-586.

[12] Moizumi K, Cybenko G. The Travelling Agent Problem[D]. Hanover, UK: Dartmouth College, 1998.

[13] Barhen N S V, Iyenger J J, Vaishnavi S S. On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(6): 740-753.

[14] Wendi R H, Anantha C, Hari B. Energy-efficient Communication Protocol for Wireless Microsensor Networks[C]//Proc. of the 33rd International Conference on System Sciences. Hawaii, USA: [s. n.], 2000.

编辑　金胡考