改进的蚁群聚类算法在本体知识库中的应用①

Applocation of Improved Ant Colony Clustering Algorithm to Knowledge Base Based on Ontology

李　秋　王建维　魏小鹏

(大连大学 先进设计与智能计算省部共建教育部重点实验室 辽宁 大连 116622)

摘　要： 为了解决基于本体的知识库构建效率不高的问题，将模块化思想引入本体中，并在蚁群聚类算法的基础上，融入遗传算法的种群思想，提出了一种基于改进的蚁群聚类算法的模块化本体的方法。以本体概念间的相似度作为划分模块的主要依据，然后应用改进的蚁群聚类算法得到本体模块划分方案，最后通过实例验证了该方法的有效性。

关键词： 遗传算法 蚁群聚类算法 模块化 语义相似度 本体 知识库

1　引言

本体(Ontology)作为一种在语义和知识层次上描述信息系统的概念模型建模工具，为用户应用共享领域知识提供了桥梁，已经广泛应用于知识工程、系统建模、信息处理、语义 Web 等领域[1]。尽管目前本体的应用越来越多，但是本体的粒度控制研究却被忽视，过大的粒度给本体的重用和维护带来了问题。众所周知,本体的构建需要耗费大量的人力和物力，尤其在某些复杂的领域,本体的体积非常巨大，本体中包含了成千上万的概念和关系，对于某个概念的某个细微变更就可能影响到整个本体，这样的本体维护起来十分困难。另一方面，在本体重用时，往往并不会对整个本体感兴趣，而只是想重用其中的一部分。但由于系统的高度耦合性，只能将整个本体全部重用，这不仅增大了开销，也降低了效率[2]。很多的研究表明本体模块化是很好的解决方法。

模块化原理在机械工程、软件工程等诸多成熟领域已经得到广泛应用并取得了成功，它能够有效地分解复杂的知识体系，因此在知识组织的过程中，可采用模块化方法构造本体以降低知识组织的复杂性[3]。将模块化思想引入本体中，有助于改善本体的建设，更有利于本体知识库的共享、重用和维护。模块划分是一个复杂的优化过程，可以使用现代优化方法对其进行优化，蚁群算法和遗传算法都属于现代优化方法。于是本文基于蚁群聚类算法，融入遗传算法的种群思想，提出了一种基于改进的蚁群聚类算法的模块化本体的方法，该方法结合蚁群聚类算法具有自组织性、可扩展性、健壮性及易与其他方法结合等特征和遗传算法对种群操作效率较高的特点，具有较高的实现效率和较强的搜索能力。从而提高了基于本体的知识库的建设和使用效率。

2　基于本体的语义相似度的计算

概念语义相似度计算在信息检索、信息推荐和过虑、数据挖掘、机器翻译等领域有着广泛的应用，成为当今信息技术研究的一个热点[4]。

力求本体概念间的相似度计算更加客观准确，结合前人研究经验，本文在计算相似度时遵循了几个原则：(1)量化原则：相似度是一个数值，一般取值范围在[0,1]之间。并且一个概念与其本身的相似度为 1，不相关的两个概念间的相似度为 0。(2)简单性原则：相似度的计算应该尽量简单以方便计算且通俗易懂。

① 基金项目:国家自然科学基金(50575026,50805010)；辽宁省高校科研计划(2008S007)；大连市优秀青年科技人才基金(2008J23 JH028)；辽宁省智能信息处理重点实验室开放课题资助项目

收稿时间:2009-02-16

(3)对称性。即 Sim(x,y)=Sim(y,x)，其中 Sim(x,y)表示 x 和 y 这两个术语之间的相似度。基于以下思想：在本体概念模型的层次结构中，概念间的语义相似度与语义距离成反比例关系，同时越靠近底层的概念所描述的信息越具体，因此若概念间最近共同祖先的深度越大，则概念的语义相似度越大；若概念之间的层次差增加，则相似度减少；并且随着概念的语义重合度增加，相似度也增加。本文提出了一种改进的基于本体的语义相似度计算方法。

本体是将某个应用领域抽象概括成一组概念及概念之间的关系，在本体的层次结构中，每个概念或概念之间的关系可以简称为“结点”。本体中，一个概念的含义可被它的子概念继承并共享，所有子概念组合起来，能够更好地描述概念的含义，这里赋予每个子概念一个权值，代表它们包含的信息量，如果一个概念的子概念越多，它的共享程度就越高，那么它的内容就越抽象，它所包含的信息量就越少，所以若记 APS(i)[5]为结点 i 的权值，那么它的运算公式为

$$APS(i)=\frac{1}{n+2},n\text{为结点}i\text{的子孙结点的数量}$$

在仙农始创的信息论中，任意随机事件的自信息量，定义为该事件发生概率的对数的负值[6]。引申到本体中，那么结点 i 的自信息可用−lgAPS(i)来表示。

设 c_1 和 c_2 是本体中的两个概念。$Sim(c_1, c_2)$表示这两个术语之间的相似度，则本文将改进的基于本体的语义相似度计算公式定义如下

$$Sim(c_1,c_2)=\frac{2\times\sum_{i=1}^{n}\delta_i(c_1,c_2)\times H_n}{Dist(c_1,c_2)\times(|H_{c_1}-H_{c_2}|+1)} \quad (1)$$

其中

$$\delta_i(c_1,c_2)=\begin{cases}-\lg APS(i) & \text{当}c_1\text{和}c_2\text{前}i\text{个父类相同时}\\ 0 & \text{当}c_1\text{和}c_2\text{前}i\text{个父类不同时}\end{cases}$$

$Dist(c_1,c_2)$为两个概念之间的距离，这里将连接他们最短路径上 n 个结点的权值之和$=\sum_{j=1}^{n}(-\lg APS(j))$

作为两个概念之间的距离；H_n 为两个概念 c_1 和 c_2 的最近共同祖先的深度，Hc_1 和 Hc_2 分别为概念 c_1 和 c_2 的深度，则(1)式可写为

$$Sim(c_1,c_2)=\frac{2\times\sum_{i=1}^{n}(-\lg APS(i))\times H_n}{(-\lg APS(c_1)-\lg APS(c_2)+\sum_{k=1}^{n}(-\lg APS(k)))\times(|H_{c_1}-H_{c_2}|+1)} \quad (2)$$

其中 K 为最短路径上除 c_1 和 c_2 之外的结点。

(2) 式中用概念间共同祖先概念的权值运算来计算语义重合度，用最短路径上结点权值运算来计算语义距离，并且考虑了深度因素的影响。无论是在语义重合度的计算中，还是计算语义距离时，都加进了概念的属性这个元素，通过属性来区别对待不同的概念。于是通过式(2)可以得到符合前面提出要求的相似度计算结果，下一步所做的工作，是将计算结果进一步进行归一化处理，即让相似度的取值范围在[0,1]之间。本文使用的归一化公式[7]如式(3)所示：

$$Sim(x,y)=1-\mu_0^{Sim(x,y)} \quad (3)$$

式中，μ_0 的取值范围为[0,1]。

3 基于改进的蚁群聚类算法的本体模块划分

3.1 遗传算法与蚁群聚类算法概述

遗传算法(Genetic Algorithms，简称 GA)最初是由美国的 J. Holland 教授于 1975 年在他的专著《自然界和人工系统的适应性》中提出的[8]。遗传算法类似于自然进化，通过作用于染色体上的基因寻找好的染色体来求解问题。它将问题域中的可能解看作是群体的一个个体或染色体，并将每一个体编码成符号串形式，模拟达尔文的遗传选择和自然淘汰的生物进化过程，对群体反复进行基于遗传学的操作(选择、交叉和变异)，根据预定的目标适应度函数对每个个体进行评价，依据适者生存，优胜劣汰的进化规则，淘汰低适应度的个体，选择高适应度的个体参加遗传操作，经过遗传操作后的个体集合形成下一代新的种群，然后对这个新种群进行下一轮进化，不断得到更优的群体，同时以全局并行搜索方式来搜索优化群体中的最优个体，求得满足要求的最优解[9]。

遗传算法是一种在很多类型的问题中非常有效的全局寻优的优化技术。它可以搜索空间的全局最优解而不必考虑局部解，除了目标函数外不必具备任何特定的知识，具有鲁棒性、隐含并行性和全局搜索等特点，因此很容易与其他技术结合，已被广泛应用到很多领域[10]。

最早的蚁群聚类算法由 Deneubourg 提出，Lumer 等首先改进此算法，提出了标准蚁群聚类算法[11]，其原理为：

定义待聚类的数据 xi 与其邻域数据的平均相似性为

$$f(x_i)=\max\{0,\frac{1}{s^2}\sum_{x_j}(1-\frac{d(x_i,x_j)}{\alpha})\}$$

式中 $d(x_i,x_j)$表示数据 x_i 和 x_j 在属性空间中的距离；S^2 是 x_i 周围邻域的表格数，从而邻域半径 $r=\frac{s-1}{2}$；是相异性常数。

数据 x_i 的拾起概率和放下概率分别定义为

$$P_{pick}(x_i)=\left(\frac{k^+}{k^++f(x_i)}\right)^2$$

$$P_{drop}(x_i)=\left(\frac{f(x_i)}{k^-+f(x_i)}\right)$$

其中，k^+、k^-是两个参数，Deneubourg 将它们分别设置为 0.1 和 0.3。

蚁群聚类算法的基本思想是将数据随机均匀散布在二维表格中，每个表格至多容纳一个，然后每个蚂蚁随机选择一个数据，根据该数据在局部邻域的相似性得到的概率，决定蚂蚁是否拾起、移动或放下该数据。经过有限次迭代，表格内的数据按其相似性而聚集，最后得到聚类结果和聚类数目。蚁群聚类方法具有许多特性，如灵活性、健壮性、分布性和自组织性等，这些特性使其非常适合本质上是分布、动态的问题求解，在解决无监督的聚类问题方面，具有广阔的前景。

该算法实际上是一种基于网格和密度的聚类方法。为了便于处理高维数据空间。首先将其映射到某一低维网格空间，映射要确保簇(将相似的对象聚合在一起形成簇)内距离小于簇间距离，同时网格的精细度将会影响聚类质量[12]。虽然标准蚁群聚类算法可以取得较好的聚类结果,但需要设置大量参数,并且当数据集规模增加时，数据的二维空间增大，聚类速度慢，于是导致计算时间长，运行效率不高等问题。

本文在此基础上进行改进，将遗传算法的种群思想融入蚁群聚类算法，将待聚类的高维数据空间映射到线性空间，即染色体，并改变了拾起和放下数据的规则，减少了参数设置。于是加快了聚类速度，提高了聚类性能。

3.2 算法描述

本体中元素之间相似度值(即关联值)的形成是本文模块划分的前提，通过第二部分提出的基于本体的语义相似度的计算方法可以得到本体中元素之间的相似度值，有了这些相似度值后，进而就可以构成本体元素间的关联矩阵。基于改进的蚁群聚类算法对本体进行模块划分的步骤如下：

step1 算法初始化，初始化蚂蚁个数、状态及位置，将所有待聚类的本体元素映射到一个染色体上。读入关联矩阵值，生成初始染色体，具体编码过程如下[13]：

(1) 假设待模块化的本体包含 n 个元素，分析关联矩阵是否有单一模块(即由一个元素组成的模块，某一元素和其他元素的关联都非常弱或非常强，则此元素应单独形成一个模块)，如果有且数目为 SM，则将其单独列出，剩下的元素数为 n=n-SM。

(2) 计算默认模块数目 $MM=[\sqrt{n}]$ (表示不大于 $\sqrt{n}$ 的最大整数)，也可以由用户自行设定模块数目。

(3) 编制最初染色体代码，本算法中的染色体采用自然数编码，规则如下：首先染色体的第 1 位为 0；然后依次插入从 1 到 n 的 n 个自然数，代表 n 个元素的代号，染色体末端再插入 0。

(4) 计算 0 插入位 Insertsite=n/MM，如果 Insertsite 可以整除，则在从 1 到 n 的 n 个自然数中，每隔 Insertsite 个位数后插入一个 0，一共插入(MM-1)个 0；如果 Insertsite 不能整除，则在从 1 到 n 的 n 个自然数中，每隔(Insertsite+1)个位数后插入一个 0，同样一共插入(MM-1)个 0，完成了初始染色体的编码工作。

step2 位于当前模块中的蚂蚁根据搬运策略交换最初染色体的基因，生成一个新的染色体，将其设为当前染色体，通过公式(7)计算该染色体的目标函数值，若目标函数收敛，则输出模块划分方案；否则进入步骤 3 执行。以下是几个相关的定义：

定义 1. 相关度的局部密度函数。

假设 R(i,j)为本体中元素 i 和 j 间的语义相似度，t 时刻蚂蚁在 r 处，且元素 i 就在当前位置。与 i 相关的局部密度函数 f(i)为

$$f(i)=(\sum_{i,j\in M(r)}R(i,j))/Nm_i \qquad (4)$$

f(i)是 i 与周围其他元素的相关密度函数；M(r)表示位置 r 所属的模块区域；Nm_i 为 i 所在模块中元素的数目。

定义 2. 拾起概率和放下概率。

聚类过程中，蚂蚁总是拾起与领域元素最不相关的个体，然后把它放到与领域元素相关度最大的位置

上[14]，拾起概率 P_u 和放下概率 P_d 定义如下

$$P_u=\begin{cases}1, f(i)=min(f(i)) 且 |max(f(i))-min(f(i))|>0.1, i\in M(r)\\ 0, \quad |max(f(i))-min(f(i))|\le 0.1, i\in M(r)\end{cases} \tag{5}$$

$$P_d=\begin{cases}1, & f'(i)=max(f'(i)) 且 f'(i)>f(i)\\ 0, & f'(i)\le f(i)\end{cases} \tag{6}$$

其中，$f'(i)=(\sum_{i\in M(r),j\notin M(r)} R(i,j))/Nm_j$，$Nm_j$ 为 j 所在模块中元素的数目。

定义 3. 目标函数。

本文使用如(7)所示的式子作为模块划分目标函数。

$$F=\sum_{m=1}^{V}\frac{fm}{fm\,max+1} \tag{7}$$

其中，$fm=\sum_{i=1}^{Nm-1}\sum_{j=i+1}^{Nm} R(i,j)$，fm 表示第 m 个模块内所有元素两两之间关联值之和，如果 R(i,j)=1，则

$$fm\,max=\sum_{i=1}^{Nm-1}\sum_{j=i+1}^{Nm}1=\frac{Nm(Nm-1)}{2} \tag{8}$$

其中，V 为模块数，Nm 为第 m 个模块中所包含的元素数。随着聚类过程的进行，目标函数 F 将逐渐变大，并趋于最大值，在设定实验停止的准则时，可以采用迭代次数或根据目标函数在一定次数内没有变化作为依据。

蚂蚁的搬运策略如下：

① 如果蚂蚁空载并在当前模块位置发现元素 i(i 属于本体概念集合)，则通过公式(4)(5)计算 f(i)和拾起概率 P_u，若 P_u=1 则拾起元素 i，蚂蚁的状态变为运载，进入蚂蚁的搬运策略②，否则转步骤 3。

② 如果蚂蚁携带元素 i，即处于运载状态，则通过公式(4)(6)计算 f(i)和放下概率 P_d，若 P_d=1 则放下元素 i，蚂蚁的状态变为空载。

step3 选择未被其他蚂蚁遍历的下一个模块的元素作为下一站。转步骤 2 执行。

4 实例分析

实验的数据来自于一个简单的电脑本体，它由 20 个概念及其关系组成，如图 1 所示。为后边计算方便，将每个概念都标了编号，其中主备选表示主机可选部件，外备选表示外设可选部件，在这里基于这个本体进行实验，利用本文提出的基于本体的语义相似度方法计算相似度，并根据目前电脑本体的规模取 $\mu=0.6$ 是比较合适的，最终得到图 2 所示关系矩阵，由于实际应用中电脑本体两个概念间的语义相似度具有对称性，图 2 中的上三角中的数据值与下三角中的值是围绕对角线对称的，所以在图 2 中只列出下三角中的数据。

采用 3.2 节的算法，首先得到初始染色体(0，2，3，4，5，6，0，7，8，9，10，11，0，12，13，14，15，16，0，17，18，19，20，0)，两个 0 之间的元素构成一个模块，由于 1 号元素和其他元素的关联都非常弱(小于 0.2)，如图 2 所示，则此元素单独形成一个模块。在 Windows 环境下，利用 Java 语言编程实现电脑本体的模块划分。最终得到模块的具体划分方案为 {1}、{4,15,16}、{2,7,8,9,10}、{3,12,13,14}、{11,17,18,19,20}，最优目标函数值是 4.2898。

从上面实例的结果看，本文采用的基于改进的蚁群聚类算法的本体模块划分方法可以准确得出本体模块划分的最优结果，对照电脑本体可看出最后得到的模块划分的方案完全符合领域专家的经验，并且该结果与使用文献[15]中提出的模块划分方法形成的优化模块划分方案相似，而且比其划分效率高，从而验证了本文方法的有效性。

图 1　电脑本体

5 结语

在基于本体的知识库建设中，本体的构建是基础。但目前本体的建设效率不高，尤其本体粒度过大造成难以重用和维护等问题越来越突出，针对这一现状，本文提出一种基于改进的蚁群聚类算法的模块化本体的方法，该方法通过计算本体概念间的语义相似度值

来建立关系矩阵，并以此作为划分模块的依据，在蚁群聚类算法中融入遗传算法的种群思想，最后得到模块划分方案，从而减少了对本体引用和维护的代价，实现了本体构建的智能化，提高了基于本体的知识库的构建效率和质量。

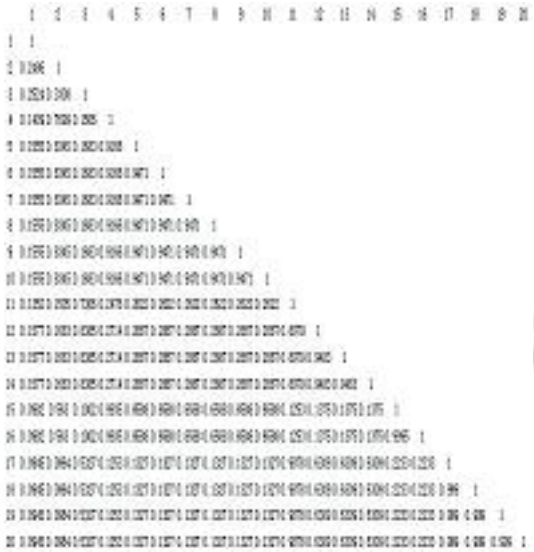

图 2 电脑本体概念间的语义相似度关系矩阵(μ=0.6)

参考文献

1 李健康,张春辉.本体研究及应用进展.图书馆论坛, 2004,24(6):80－86.

2 张维一,陆汝占.本体模块化的研究与实现.计算机应用研究, 2007,24(11):206－209.

3 冯兰萍,朱礼军,张继国.一种基于模块化本体的知识组织方法研究.现代图书情报技术, 2007,12:30－33.

4 黄果,周竹荣.基于领域本体的概念语义相似度计算研究.计算机工程与设计, 2007,28(10):2460－2463.

5 Schickel-Zuber V, Faltings B. OSS: A Semantic Similarity Function based on Hierarchical Ontologies. International Joint Conferences on Artificial Intelligence (IJCAI2007),Hyderabad, India: 2007:551－556.

6 李珊,何建敏,厉浩.基于本体和加权互信息的专业知识检索.情报学报, 2006,25(5):559－563.

7 李鹏,陶兰,王弼佐.一种改进的本体语义相似度计算及其应用.计算机工程与设计, 2007,28(1):227－229.

8 Holland JH. Adaptation in natural and artificial systems.Michigan: University of Michigan Press, 1975.

9 郭清华.遗传算法概述.中国教育教学杂志(高等教育版), 2006,12(143):81－82.

10 何明.基于遗传算法和粗糙集理论的增量式规则获取方法.西安石油大学学报(自然科学版), 2008, 23(4):101－105.

11 Lumer E, Faieta B. Dibersity and adaptation in populations of clustering ants. Proc. Third International Conference on Simulation of Adaptive Behavior: From Animals to Animates 3. Cambridge, MA: MIT Press, 1994:91－95.

12 张建华,江贺,张宪超.蚁群聚类算法综述.计算机工程与应用, 2006,16:171－174.

13 单泉,闫光荣,雷毅.改进模拟退火算法在模块划分中的研究及应用.计算机工程, 2007,33(12):208－210.

14 邓可,林杰.基于蚁群聚类算法的大规模定制产品模块划分研究.计算机工程与应用, 2008,44(2):130－132.

15 林松涛.模块化本体建设研究[博士学位论文].北京:北京邮电大学, 2006.