用子空间粒子群聚类算法识别 Folksonomy 标签冗余的研究

王晓帅 覃 华 丁立朵 马翩翩
(广西大学计算机与电子信息学院 南宁 530004)

摘 要 Web2.0标签系统中经常包含很多冗余的标签,标签冗余会增加用户选择喜好项目时的负担,从而影响用户建模和对推荐系统的评估。标签数据集通常存在着大量不相关或是冗余的特征,而不同簇之间的相关特征子集又是不一样的,所以应该从不同的特征子集中来发现簇。提出使用子空间粒子群聚类识别标签冗余,算法采用指数型变权类似 K-means 的目标函数,该函数对变量权值的改变更加敏感。在此基础上利用粒子群优化目标函数搜寻得到全局最优的标签聚类,提高抽取冗余标签的准确度。实验结果表明,此算法具有较强的全局搜索能力,应用于标签冗余识别获得了更好的精度。

关键词 Web2.0标签推荐系统,标签冗余,子空间粒子群聚类
中图法分类号 TP393 **文献标识码** A

Particle Swarm Optimization for Subspace Clustering Identify Tag Redundancy in Folksonomy

WANG Xiao-shuai QIN Hua DING Li-duo MA Pian-pian
(Dept. of Computer and Electronics Information, Guangxi University, Nanning 530004, China)

Abstract Web2.0 tag recommender systems always contain a lot of redundant special label. Redundancy on tags may even burden the user with additional effort by selecting their preferred items, and these redundancy can introduce erroneous features into the user profile and hamper the effort to judge recommendations. There is usually a lot of irrelevant or redundant features in high-dimensional data sets, feature subsets between different clusters are not the same. Therefore, we should focus on the different features subsets to discover the cluster. This paper proposed subspace PSO clustering to identify tag redundancy. We designed a suitable weighting K-means objective function, which is more sensitive to the change variables in weight. On this basis we developed PSO to optimize the objective function, then obtain global optimal value, and finally improve tag redundancy accuracy. Our experimental results show that the proposed algorithm has greater searching capability and obtains a better clustering accuracy.

Keywords Web2.0 tag recommender systems, Tag redundancy, Subspace PSO clustering

1 引言

在Web2.0的标签推荐系统中,自定义标签形式的 Folksonomy 在时下的社会性网络服务中得到广泛的应用。Folksonomy 包含3个基本元素:资源、用户和标签[1-3],可以用四元组 D 来表示:$D=\langle U,R,T,A\rangle$。其中 U 是用户集合,R 是资源集合,T 是标签集合,A 是标注关系集合。在 Folksonomy 中,用户可以自由地为资源添加"标签"。每个标签相当于用户对资源的一个分类,资源根据不同的标签很容易地被分类、排序和检索,用以上任何两个组成部分,用户都可以发现具有相似偏好的其他用户,以及相似的资源或已用的相似标签。尽管 Folksonomy 提供了诸多的好处,其也面临着一些独特的挑战。用户根据个人的使用习惯,以自定义的自由词做标签,导致了标签的冗余。这些冗余标签加重了用户在选择喜好项目时的负担,同时也影响了对 Folksonomy 标签推荐系统的评估质量[4]。针对标签冗余问题,Gemmell 等人[4]利用 K-means 聚类识别冗余标签,然后剔除冗余度较高的标签,并且将冗余标签提供给评估系统以便做出正确的评估。张新伦等[6]利用核 K-means 聚类对冗余标签进行聚类,以达到抽取冗余标签的目的。

2 子空间聚类

标签数据是高维的,对高维的数据进行聚类比较困难,而且标签数据分布稀疏,其数据间的距离几乎相等,因此用传统的聚类算法处理如此高维的标签数据时有效性大为降低。对一个簇而言,通常存在着大量不相关或是冗余的特征;而不同簇之间的相关特征子集又是不一样的。因而,高维数据聚类更应该是在不同特征子空间发现簇,而不是在所有维度空间去寻找簇。上述的聚类方法称为子空间聚类,它们尝试在不同子空间发现聚类。目前有多种类型的子空间聚类算法,它

本文受国家自然科学基金项目(61063032),教育部人文社会科学研究项目(11YJAZH080)资助。
王晓帅(1985-),女,硕士生,主要研究方向为电子商务技术与应用、数据挖掘等,E-mail:wxsh163@126.com;覃 华(1972-),男,副教授,主要研究方向为电子商务数据挖掘、最优化技术等;丁立朵(1986-),女,硕士生,主要研究方向为电子商务技术与应用、数据挖掘等;马翩翩(1986-),女,硕士生,主要研究方向为数据挖掘、个性化推荐等。

们根据加权方式的差异来衡量数据对象间的相似性。其中，软投影聚类获得了广泛的关注。其核心思想是，通过迭代来最小化某一个目标函数，并且数据对象之间的距离是在整个维度空间进行的，但是该维度空间是被加权的。变量权值转换了距离，使得高维空间中聚类中的数据对象被重塑成密度高的超球体，因而容易被聚类算法有效识别。软投影聚类算法依赖于搜索策略和评测标准来驱动。因此，如何定义一个有效的目标函数以及如何高效地搜索到最优变量权值是软投影聚类的两大问题。

近年来，一些软投影聚类算法已被相继用来识别聚类，2005 年 J. Z. Huang 等人[7]提出的 W-k-means 算法为每一维分配一个权值，尽量在加权的变量空间里最小化聚类内部距离的累加，但其变量权值的更新依赖于参数 β 的数值。此外，W-k-means 算法没有使用一个有效的局部搜索策略，所以通常不能识别嵌在变量子空间的聚类，因而不适合聚类高维数据。2007 年 Domeniconi C 等人[8]提出 LAC 算法，其变量加权的更新主要依赖参数 h，不能根据数据集及其各异的簇结构特点进行自动调节。2007 年 LipingJing 等人[9]提出了 EWKM 算法，该算法引入了维度“权重熵”因子，其目标是令投影子空间各维度权重分布具有最大的熵，这基于维度权重服从某种概率分布的假设，但是该算法还存在如何在权重熵因子和扩展的 K-means 优化目标之间取得平衡的问题。另外，这 3 种算法都是源于 K-means 聚类算法，它们对初始簇中心都很敏感。更重要的是，它们采用的是局部搜索策略最小化目标函数，通过迭代加快了收敛速度但是聚类质量大为下降。因此，需要一种有效的搜索方法来加快收敛。

从根本上说，子空间聚类算法的变量加权是一种带约束的连续非线性优化问题。而粒子群优化在解决复杂的参数评估上是一种很好的优化方法，因此我们提出用粒子群优化来处理子空间在聚高维标签数据上的变量加权问题。首先为每个簇类的每个变量分配一个权值，接着采用特殊的加权 K-means 函数，它更适合在每个簇类自己的子空间上计算簇类内的距离和。其次，利用了一个简单的非规格化的权重表示方法来将带约束的搜索空间改造为冗余的封闭空间，使搜索变得更容易。最后，不采用局部搜索策略，而是利用粒子群优化来最小化给定的目标函数。

3 子空间粒子群聚类算法

我们的算法在两个种群，即位置种群[10-12]和个体最优位置种群[10-12]上执行变量加权。在变量加权空间中，位置种群是搜索的基础，位置种群的进化注重的是全局搜索，而个体最优位置种群追随最优适应值的位置，加快收敛速度。此外，通过个体最优位置的相互交叉学习来维持种群的多样性。

3.1 子空间粒子群优化算法

与其他进化寻优算法类似，子空间粒子群优化算法通过个体间的协作与竞争，实现多维空间中最优解的搜索。子空间粒子群优化算法初始化为一群随机粒子(随机解)，在 m 维搜索空间中，第 i 个粒子在 m 维空间中的位置表示为 $x_i=(x_{i1},x_{i2},\cdots,x_{im})$，第 i 个粒子经历过的最好位置记为 $P_i=(p_{i1},p_{i2},\cdots,p_{im})$，每个粒子的飞行速度为 $V_i=(v_{i1},v_{i2},\cdots,v_{im})$，$i=1,2,\cdots,s$。在整个群体中，所有粒子经历过的最好位置为 $pBest=(p_{g1},p_{g2},\cdots,p_{gm})$，粒子间通过交叉学习获得 $CpBest$，维持种群的多样性并引导粒子向最优点靠近，最终获得全局最优位置 $gBest$，每一代粒子基于下面公式更新的速度和位置：

$$X_i(t+1)=V_i(t+1)+X_i(t) \tag{1}$$

$$V_i(t+1)=\alpha\cdot V_i(t)+c_1\cdot r_1\cdot(CpBest_i-X_i) \tag{2}$$

在该算法中，我们需要 3 个种群：一是变权位置种群 W，其数值分布在一定的范围 R 内；二是变权速度种群 V，其数值分布在范围$[-\max v,\max v]$。其中 $\max v$ 表示飞行的最大速度，$\max v=0.25*R$；三是簇中心点种群 Z。它们都是 $k*m$ 的矩阵。其中，α 为惯性权重；c_1 为加速因子；r_1 为随机数；$CpBest_i$ 是有粒子们的个体最优位置相互学习得到的。通过式(1)和式(2)，本算法能够维持种群的多样性并避免过早收敛。

3.2 粒子编码

粒子群算法采用实数编码，一个编码对应于一个可行解，这种方法有利于改善粒子群算法的计算复杂性，提高运算效率。由于被优化的问题的解是找到最优的聚类中心，因此本文采用的是基于聚类中心的编码方式，也就是每个粒子的位置是由 m 个聚类中心组成，粒子的长度是 $k\times m$，m 为聚类中心的维数。也就是说 k 个聚类中心顺序排列构成一个粒子，粒子的组成表示了其在空间中的位置。

3.3 目标函数

在子空间粒子群聚类算法中我们最小化如下的目标函数[3]：

$$F(w)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{l,j}\cdot\left(\frac{w_{l,j}}{\sum_{j=1}^{m}w_{l,j}}\right)^{\beta}\cdot d(x_{i,j},z_{l,j})$$

$$\text{s.t.}\begin{cases}0\leqslant w_{l,j}\leqslant 1\\ \sum_{l=1}^{k}u_{i,l}=1,u_{i,l}\in\{0,1\},\ 1\leqslant i\leqslant n\end{cases} \tag{3}$$

式中，n,k,m 分别代表数据对象的数目、簇的数目和数据集的维度。$u_{l,j}$ 表示数据集 i 分配到簇 l 的隶属度；$w_{l,j}$ 表示簇 l 的数据的第 j 维权值；$x_{i,j}$ 表示数据 i 第 j 维的数值；$z_{l,j}$ 代表簇 l 的第 j 维的中心值；d 用来衡量两个数据对象的相异性。

上述的目标函数实际上是一个泛化的目标函数。$\beta=0$ 时，它与 K-means 的目标函数类似，不同点在于变权和约束的表示上。$\beta=1$ 时，它又与 EWKM 目标函数的前半部分类似，不同点也在于变权和约束的表示上。如果 $w_{l,j}=w_j\ \forall l$，则与 W-k-means 的目标函数类似，区别是 W-k-means 用单一的变权向量表示。不同点也在于变权和约束的表示上。在子空间粒子群聚类算法中，β 是用户定义的参数，它的取值为非负值。建议 β 取一个较大的值(通常约为 10 左右)。β 值大一些，目标函数会对权重值的变化更敏感。随着簇间差异性的权重值的变大，它会放大这些变量的影响。在区分相关维与非相关维上，它将发挥重要的作用。

在现存的软投影聚类算法中，$k*m$ 的变量加权矩阵 W 中的每个元素是分配到每一个簇的每一维的权重值，是一个实数，权重值通常是目标函数的解。它应当同时满足每行的权重值之和规格化为 1。然而，随着簇的数目越多，约束的数目也会变得越多，这就很难优化其对应的目标函数。在子空间粒子群聚类算法中，采用非规格化的矩阵 W。虽然对于带约束的目标函数的非规格化表示使得搜索空间是冗余的，但是它可以在不影响最终结果的基础上放宽约束。因此，优化

过程不再需要处理很多的约束，使优化过程变得非常简单。

3.4 子空间粒子群聚类算法描述

用 K 个聚类中心 $X=(m_{11},m_{12},\cdots,m_{1m};m_{21},m_{22},\cdots,m_{2m};\cdots;m_{k1},m_{k2},\cdots,m_{km})$ 作为聚类问题的解。解聚类问题的算法描述如下：

设定粒子数 s，规定迭代次数 max，随机产生 s 个初始解 X_0；

(1)根据当前位置，以式(3)计算目标函数值 F_0，当前适应值为个体极值 $pBest$，当前位置为个体极值位置 $pxBest$，根据各个粒子的个体极值 $pBest$，经交叉学习得到 $CpBest$，找到全局极值 $gBest$ 和全局极值位置 $gxBest$；

While(迭代次数＜规定迭代次数 max 或者目标函数取得最小值)do

for $j=1:s$

(2)按式(2)更新自己的速度，并把它限制在 max v 内；

(3)按式(1)更新当前的位置；

(4)根据当前位置，各个样本按最小聚类原则(4)分配给 k 个聚类中心；

$$\sum_{j=1}^{m}\left(\frac{w_{l,j}}{\sum_{j=1}^{m}w_{l,j}}\right)^{\beta}\cdot d(z_{l,j},x_{i,j})\leqslant\sum_{j=1}^{m}\left(\frac{w_{q,j}}{\sum_{j=1}^{m}w_{q,j}}\right)^{\beta}\cdot d(z_{t,j},x_{i,j}) \tag{4}$$

数据分配到最近的簇后，将相应隶属度矩阵中的 $u_{l,j}$ 记为 1；并对新得到数据成员的簇，按照式(5)重新计算聚类中心；

$$z_{l,j}=\frac{\sum_{i=1}^{n}u_{l,j}\cdot x_{i,j}}{\sum_{i=1}^{n}u_{l,i}},1\leqslant l\leqslant k\text{ 并且 }1\leqslant j\leqslant m \tag{5}$$

(5)计算计算目标函数值 F_1；

(6)如果 $F_1(j)<pBest(j)$，则 $pxBest(j)=X_1(j)$，$pBest(j)=F_1(j)$

End；

(7)根据各个粒子的个体极值 $pBest$，找出全局极值 $gBest$ 和全局极值位置 $gxBest$；

(8)$X_0\leftarrow X_1$

End；

(9)最后输出全局极值 $gBest$ 和全局极值位置 $gxBest$，全局最优位置即为最优的聚类中心。

4 子空间粒子群聚类识别冗余标签

在 Folksonomy 中，标签冗余指的是多个标签有同一种含义。例如大小写“Java”or “java”；词的不同形式“blogs ”or “blogging”；拼写错误“photo”or “fhoto”；缩写字等等。冗余标签会影响对推荐系统的评价质量，如下面的例子：测试集中给出的标签是“RecSys”，而经过推荐在支持集中给出的标签是“rec_sys”，此时评估系统认为这个推荐是错误的，结果低估了推荐系统的性能。评估系统若可以识别冗余标签，就可以避免上述的情况。本文把在同一个标签类中出现的标签视为冗余标签[4]，标签之间的冗余度是由标签类的冗余度体现的，一个标签类的冗余度越高，说明该类内的标签越相似，单个标签类冗余度的计算公式[6]如下：

$$r(c)=\left(\frac{f(s)-cf(c,s)}{f(s)}\right) \tag{6}$$

式中，$cf(c,s)$ 表示标签类内不重复的资源数，$f(s)$ 表示类内标记过的资源数。

所有标签类的平均冗余度计算公式[7]如下：

$$r(c*)=\sum_{c\in c_r}\left(\frac{f(s)-cf(c,s)}{f(s)}\cdot\frac{n_c}{N}\right) \tag{7}$$

式中，nc 表示各个标签类内的标签数，N 表示数据集中总的标签数。

标签类中的冗余度越高，标签的聚类结果越好，也就是将含义最相近的标签聚集到一起，这样可以给推荐系统提供更准确的冗余标签。

用子空间粒子群聚类识别冗余标签的算法描述如下：

设定种群规模，学习因子，惯性权重，加速常量，交叉学习概率等参数的数值。

输入：簇的数目，数据集。

输出：标签类的冗余度，冗余标签。

Step1 用子空间粒子群聚类算法对标签进行聚类。

Step2 根据标签聚类结果，计算每个标签类内不重复的资源数 cf(c,s)以及在该类内所有标注过的资源数 f(s)，并得到标签总数 N 和类内标签数。

Step3 根据 Step2 所得结果，利用式(6)计算单个标签类的冗余度，利用式(7)计算所有标签类的平均冗余度。

Step4 根据标签类的冗余度确定冗余标签。

5 实验

实验平台为 PC(Pentium4，CPU 3.0GHz，内存 512M)，操作系统是 Windows XP。

5.1 实验一：验证算法可行性

为评价子空间粒子群聚类算法的聚类性能，采用 UCI 公共数据库(http://kdd.Ics.Uci.edu)提供的著名的 Iris，Breast-cancer，Vowel，Glass 数据集，这些数据集有原始分类，可用于最终的聚类性能评价。本文对所选数据集进行聚类，并将最终聚类结果与原始的粒子群聚类算法、K-means 聚类算法的结果利用精确度进行比较。

聚类精确度计算如下：

$$\text{精确度}=\frac{\sum_{l=1}^{k}N_l}{n} \tag{8}$$

式中，N_l 是在第 l 个聚类中，能够被正确识别的数据对象的数目，n 是整个数据集中数据对象的数目。聚类精确度即为可被算法正确识别的数据对象占整个数据集的比例。

表 1 算法的精确度比较

数据集	子空间粒子群聚类	PSO	K-means
Iris	98.89%	87.47%	84.36%
BC	97.38%	94.89%	95.86%
Glass	76.75%	62.41%	60.16%
Vowel	67.64%	54.65%	51.69%

从表 1 看出，对这 5 种样本数据集进行聚类，子空间粒子群聚类算法比普通粒子群聚类和 K-means 聚类算法效果好。Iris 和 Breast Cancer 两个数据集的聚类精确度很高。Glass 数据集所有特征的取值都非常集中，而 Vowel 数据集所有维的取值都非常分散，从 Glass 和 Vowel 这两种比较极端的数据集的聚类结果看，子空间粒子群聚类效果还是不错的。总体上看，该算法能够较好地区分数据集的各个维对识别不同类簇的贡献程度，验证了此算法是可行有效的。

5.2 实验二：聚高维标签冗余数据

为了测试子空间粒子群聚类算法对高维标签数据的聚类效果，本文选取 PKDD2009 提供的 BibSonomy 数据集进行验证。BibSonomy 网站是基于 Folksonomy 框架系统，本文采用的源数据集包含两个数据文件(tas，bookmark)，其中 tas 文件包含用户、tas_id、标签和对应 bookmark_id 的关系记录，bookmark 文件包含资源、资源描述、bookmark_id 和对应 tas_id 的关系记录。这两个数据文件间由 tas_id 和 bookmark_id 连接，连接后的记录总共有 24 万余条。

对标签数据聚类，并寻找出最优的聚类结果。其中子空间粒子群聚类算法最大迭代次数设为 500，进行 10 实验，比较 10 次的适应值，通过最小适应值找到最好的聚类结果。适应值的变化和收敛时间如图 1、表 2 所示。

从图 1 总体上看，10 条曲线分别呈递减趋势变化，前 200 次迭代，曲线变化不是很明显，紧接着的 50 次迭代，曲线骤然下降，并缓慢地达到最低点。从曲线的变化趋势，我们可以看出，目标函数随着迭代次数的增加，最终都能取得最小适应值，前 200 次迭代，粒子们通过跟踪两个最优值来更新自己，并处在局部最优状态，接着随着迭代次数的增加，粒子们从局部最优迅速地向全局最优粒子靠近，跳出局部最优，最终达到全局最优。比较 10 条曲线可以看出第 5 次实验适应值最小，故此次聚类效果最优。

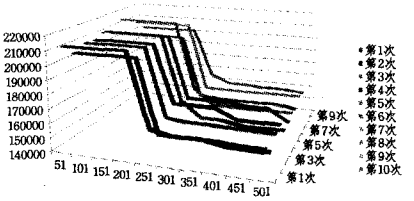

图 1 适应值随迭代次数的变化

表 2 算法的收敛性

试验次数	收敛时间	试验次数	收敛时间
1	91.156 s	6	82.14 s
2	100.985 s	7	82.219 s
3	90.063 s	8	97.266 s
4	88.859 s	9	88.39 s
5	86.063 s	10	80.844 s

表 2 记录了该算法每次实验的收敛时间，从整体看，收敛时间平均在 88.798s 左右。

在标签类的计算中，标签类的冗余度越高，此类标签越相似，说明聚类效果越好。所以说，聚类效果的好坏直接影响到标签类的冗余度。标签类的冗余度与标签类内的标签所标记的不重复资源数 $cf(c,t)$ 成反比，而标签是按照其所标记的资源来聚类的，所以标签聚类的效果越好，说明标签类中标签标记的资源越相似，也就是不重复的资源越少。

分别用子空间粒子群聚类算法和核 K-means 聚类算法对标签聚类，各做 6 次实验，每次实验迭代次数设为 500，分别取 6 次最优值。实验结果如表 3、图 2 所示。

从表 3 可以看出，采用子空间粒子群聚类识别标签冗余，标签类的平均冗余度为 0.8422，而采用核 K-means 聚类识别标签冗余，标签类的平均冗余度为 0.8026，表 3 结果表明：子空间粒子群聚类算法更能准确地抽取冗余标签。

图 2 是子空间粒子群算法和核 K-means 算法分别迭代 500 次，其收敛时间的对照图。可以看出，在不失精度的情况下，子空间粒子群算法在处理高维标签冗余数据时收敛时间较快。

表 3 标签的冗余度

实验	核 K-means 聚类	子空间粒子群聚类
1	0.7756	0.8593
2	0.8021	0.8158
3	0.8176	0.7965
4	0.7945	0.8417
5	0.8220	0.8791
6	0.8037	0.8608
均值	0.8026	0.8422

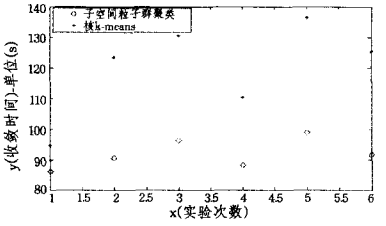

图 2 算法的收敛时间

结束语 标签数据中往往存在许多不相关的属性，使得要寻找的目标类只存在于某些低维子空间中，而不同的簇类其关联的子空间通常也是不一样的。在高维空间中我们采用子空间聚类算法挖掘隐藏在不同低维子空间中的簇类。本文提出用子空间粒子群聚类算法来处理标签冗余，从实验结果看，子空间粒子群聚类算法能够普遍获得优于传统聚类算法的聚类有效性评价指数，同时能够大大地降低传统聚类方法陷入局部最优的概率，并在一定程度上降低传统方法对初值的敏感度。算法在处理高维的冗余标签数据时，通过迭代寻找到最优适应值并获得了冗余度更高的标签类，找到了更准确的冗余标签，识别了这些冗余的标签，从而提高了推荐系统的评估质量。

参 考 文 献

[1] Zhang Ning, Zhang Yuan, Tang Jie. A Tag Recommendation System based on contents[J]. ECML PKDD Discovery Challenge 2009(DC09), 2009, 497: 285-295

[2] Zhang Yuan, Zhang Ning, Tang Jie. A Collaborative Filtering Tag Recommendation System based on Graph[J]. ECML PKDD Discovery Challenge 2009(DC09), 2009, 497: 297-306

[3] 杨丹，曹俊. 基于 Web2.0 的社会性标签推荐系统[J]. 重庆工学院学报，2008，22(7)：51-55

[4] Gemmell J, Ramezani M, Schimoler T. The Impact of Ambiguity and Redundancy on Tag Recommendation in Folksonomies[M]. NewYork, USA, ACM, 2009: 45-52

[5] Lu Yan-ping. Particle Swarm Optimizer for Variable Weighting in Clustering High-dimensional Data[J]. IEEE. Mach Learn, 2011, 82: 43-70

[6] 张新伦，苏一丹，惠刚刚. 核 K-means 聚类在 Folksonomy 标签模糊和冗余中的应用[J]. 计算机应用，2011(3)：680-682

[7] Huang J Z, Ng M, Rong H, et al. Automated dimension weighting in k-means type clustering[J]. IEEE Trans. on Pattern Analysis and Machine Intelligellce, 2005, 27(5): 1-12

[8] Domeniconi C, Gunopulos D, Ma S, et al. Locally adaptive metrics for clustering high dimensional data[J]. Data Mining and Knowledge Discovery Journal, 2007, 14: 63-97

[9] Jing L, Ng M K, Huang J Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensinoal sparese data [J]. IEEE Trans. on Knowledge and Data Engineering, 2007, 19 (8): 1026-1041

[10] 李俊金，向阳，芦英明，等. 粒子群聚类算法综述[J]. 计算机应用研究，2009，126(112)

[11] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions onEvolutionary Computation, 2006, 10(3): 281-295

[12] 陈黎飞，郭躬德，姜青山. 自适应的软子空间聚类算法[J]. Journal of Software, 2010, 11(21): 2513-2523

(上接第 278 页)

本文算法为了提高社区挖掘的精确度，在每个基础周期都会重新运行算法，并且进行新的社区划分。这一行为虽然在一定程度上提高了算法的时间复杂度，但是与其他算法相比，仍然具有时间的优势。表 1 中，T 代表算法的总运行时间，t 代表算法的基础周期，则结果如表 1 所列。

表 1 时间复杂度对比分析

算法名称	参考文献	时间复杂度
N-G 算法	[3]	$O(m^2n)$
G-N 算法	[2]	$O(n^2m)$
BB 算法	[7]	$O(n^3)$
本文算法		$O(\frac{T}{t}nlog(m+n))$

4 实验结果分析对比

本文基于现有的静态社会网络，以时间为轴模拟真实社会网络的动态形成过程。根据节点的度数分布，逆向推倒社会网络中节点的出现顺序。采用 Matlab 编程，取得如图 4 和图 5 所示的实验结果。

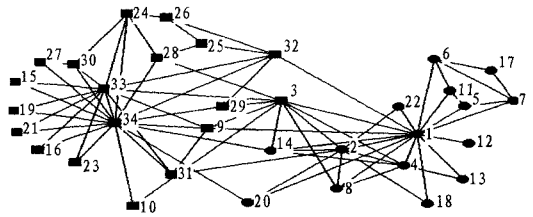

图 4 本文算法对 Zachary Club 的划分结果

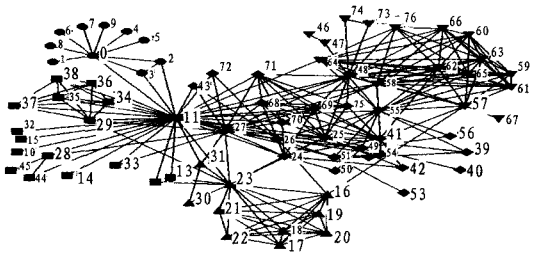

图 5 本文算法对 Les Miserables 的划分结果

根据节点的度数分布，以时间为轴分析 Les Miserables 网络的形成演化过程。

图 6 中，横轴代表时间，纵轴代表节点度数，竖轴表示具有不同度数的节点的数量。

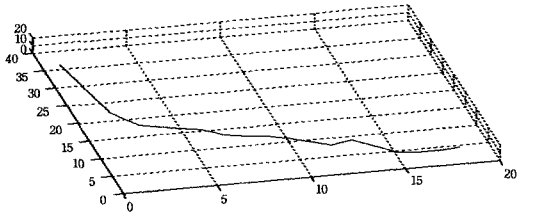

图 6 悲惨世界社会网络的数字化形成过程

从图 6 可以看出，度数最大的节点最先出现，然后陆陆续续有新的节点加入到悲惨世界数据集中，后来的节点都偏向于与度数大的节点建立链接，因此导致最先出现的节点度数的增长最快。同时，上图阐释了社会网络中节点的度数符合幂分布规律[14]。

结束语 通过对社会网络形成集演化机制的研究，本文提出了一种基于扩散理论的动态社区挖掘。利用某一时刻的静态社会网络结构模型进行逆向推理分析，模拟真实社会网络的形成；同时，不断地对新加入的节点进行关系分析和相似性度量，进而将节点划分到其所属社区中。实验指明，该算法不仅具有较高的模块度，而且能够保持较低的时间开销，大大降低了算法的复杂度，具有较好的实用价值。

参 考 文 献

[1] Dorogovtsev S N, Mends J F F. Evolution of Networks: From biological Nets to the Internet and WWW[M]. Oxford University Press, Oxford, 2003

[2] Friedkin N. A structure theory of social influence[M]. Cambridge University Press, Cambridge, 1998

[3] Barabasi A L, Albert. Emergence of scaling in random networks [R]. Science, 1999, 286(5439): 509-512

[4] 郭鸿，周娅. Web 结构挖掘中 HITS 算法的改进[J]. 信息化纵横，2009，16：70-73

[5] Madria S K. Research issues in Web data Mining[J]. Data Warehousing and Wlege discovery, 1999, 99: 303-312

[6] Bianconi G, Barabsi A. Competition and Multiscaling in Evolving Networks[J]. Europhysics Letters, 2001, 5: 436-442

[7] Newman M E J, Girvan M. Finding and Evaluating community structure in networks[R]. E69, 026113, Physics Review, 2004

[8] Kernighan B W, Lin S. A efficient heuristic procedure for partitioning graphs[J]. Bell System technical Journal, 1970, 49(2): 291-307

[9] Radicchi F, Castellano C, Cecconi F, et al. Defining and identifying communities in networks[C]// Proceedings of the National Academy of Sciences. USA, 2004: 2658-2663

[10] 高琪，张永平. PageRank 算法中主题漂移的研究[J]. 网络与通信，2010，3(3)：117-119

[11] Krapivsky P L, Render S. Connectivity of Growing Random Networks[R]. Physical Review Letters, 2000, 85: 4629-4632

[12] Wolf J, Aggarwal C, Wu K L, et al. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering [C]// Proceeding of the 5th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. 1999: 201-212

[13] 艾伯特. 巴拉巴西. 链接网络新科学[M]. 徐彬，译. 长海：湖南科学技术出版社，2007

[14] Albert R, Barabasi A L. Statistical mechanics of complex networks[R]. Reviews of Modern Physics, 2002: 74