粒度智能体进化分类算法

潘晓英[1], 焦李成[1], 刘 芳[2]
(1. 西安电子科技大学智能感知与图像理解教育部重点实验室和智能信息处理研究所, 陕西西安 710071;
2. 西安电子科技大学计算机学院, 陕西西安 710071)

摘 要: 受基于文化进化机制的粒度进化计算启发,针对分类任务提出了一种粒度智能体进化分类算法. 该方法以粒度智能体表示具有相似属性的数据;以其中包含的知识库来指导粒度智能体的进化;设计了适合分类问题的粒度进化算子——同化算子、交换算子以及分化算子,分别体现了智能体的竞争性、协同性以及自学习性. 最终根据一定的策略从所得到的粒度智能体中提取出分类规则,用以对新数据的预测分类. 测试结果表明该算法具有良好的分类预测性能,且仅需要较小的训练时间代价. 在UCI中的大部分数据集上都要优于性能良好的G-NET, OCEC以及C4.5算法.

关键词: 分类; 粒度智能体; 同化; 交换; 分化
中图分类号: TP18 文献标识码: A 文章编号: 0372-2112 (2009) 03-0628-06

Granular Agent Evolutionary Algorithm for Classification

PAN Xiao-ying[1], JIAO Li-cheng[1], LIU Fang[2]
(1. *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an, Shaanxi 710071, China;* 2. *School of Computer Science and Engineering, Xidian University, Xi'an, Shaanxi 710071, China*)

Abstract: By inspiration of the granular evolutionary algorithm, a Granular Agent Evolutionary algorithm for Classification (GAEC) is proposed. The method uses the granular agent to denote the set of examples that have similar attributions and the knowledge base guides the evolutionary of granular agent. Also some granular evolutionary operators are designed for classification problem. Assimilation operator, exchange operator, and differentiation operator reflect the competitive, cooperative and self-learning ability of agent respectively. Finally, some classification rules are extracted from granular agents by some strategies to forecast the sort of new data. Empirical studies show that the algorithm has a good classification prediction, and only need a small price for the training time. In most UCI datasets, the performance of GAEC is better than G-NET, OCEC and C4.5, which have good performance.

Key words: classification; granular agent; assimilation; exchange; differentiation

1 引言

分类算法是数据挖掘中的一个重要研究课题,其特点是根据数据集去构造一个分类器,并利用它对新数据进行分类预测. 最常用的分类算法是决策树归纳法,但是当数据集非常大时,常常由于计算量太大而无法应用. 相关的改进方法有统计和粗糙集方法、神经网络方法、贝叶斯方法等,这些算法的不足之处在于需要额外的专家知识[1]才能有效地工作. 近年来,有关学者提出了基于遗传的分类算法[2,3],并取得了较好的分类结果,但是在进化过程中可能产生退化现象. 另外,这些算法均以规则作为个体,再利用遗传算子来对其进行进化,以得到较高性能的分类规则. 这样的方式在进化过程中不可避免地会产生一些没有意义的规则. 为解决这个问题,刘静等[4]提出了自下而上的组织协同进化分类方法,该方法让进化操作直接作用于数据,最后再根据进化结果提取规则,实验结果表明该算法具有良好的分类性能. 但是在整个进化过程中,组织的选择以及组织上的操作采用了完全随机的方式,同时,对于组织的形成以及所定义的组织进化算子没有任何理论指导,这样的方式不可避免地会带来迭代次数过多,预测准确率不高等缺点.

粒度进化计算[5]是一种模拟人类社会发展和进化过程的算法,和传统遗传算法相比,它包含了两个层面

收稿日期: 2007-10-23; 修回日期: 2008-11-05
基金项目: 国家自然科学基金(No. 60703107, 60703108, 60703109, 60702062); 国家高技术研究发展计划(863计划)课题(No. 2006AA01Z107); 国家教育部博士点基金(No. 20060701007); 国家973项目(No. 2006CB705700); 教育部长江学者和创新团队支持计划(No. IRT0645); 陕西省自然科学基金(No. 2007F32)

上的进化, 一个是群进化, 即粒度的进化; 另一个为超群进化, 即以群体为单元的进化. 受其启发, 我们在粒度进化中结合了智能体的概念, 并将其用于数据挖掘的分类任务当中, 提出了一种粒度智能体进化分类算法(Granular Agent Evolutionary algorithm for Classification, GAEC).

2　粒度进化计算

文献[5]中基于文化进化的机制和特征抽象出了群进化和超群进化的概念, 并以粒度进化来表示这样的进化机制. 同时指出, 粒度进化包含了两层意思: 群进化和超群进化, 这两个进化层面不是独立分开的, 而是有机统一的, 它们的协调一致才能构成完整的粒度进化. 和一般的进化算法不同, 粒度进化模拟的是人类社会的发展和进化过程. 在整个进化过程中, 每个群体、每个个体都希望自己的利益得到最大化, 但是不断膨胀的利益获取欲望可能会侵犯别人的利益, 所以就产生了某种约束. 因此, 粒度之间协调的基本方法是: 在不侵犯别人利益的前提下, 寻求自己利益的最大化.

为完成上述的粒度进化过程, 要求粒度必须具有一定的智能性和自主性, 以完成超群进化基本功能所需要的技术支撑, 如粒度可通过与别的粒度进行协作以获得各自利益的增加, 粒度的分化、同化等功能. 因此在粒度进化中引入了智能体的概念, 将具有智能体特性的群体称之为"粒度智能体", 以 Agent 和多 Agent 的相关技术对粒度进化的原理以及实现机制进行描述和研究. 粒度智能体和多粒度智能体系统具体定义如下[5]:

定义 1　一个粒度智能体可表示为一个三元组: Agent= (P, K, G), 其中 P 表示进化群体在时刻的状态; K 表示在时刻知识库的状态; G 表示在时刻群体的目标.

定义 2　多粒度智能体系统表示为二元组(A, R), 其中 A 表示粒度智能体的集合, R 为 A 上关系的集合, 与具体问题相关, 整个二元组记为 MGAS.

3　粒度智能体进化分类算法

对于数据集中的数据, 数据的属性有可能包括连续属性, 也有可能包含一些缺失属性. 由于本文仅考虑对包含离散属性的数据进行分类, 因此对于这些情况, 必须对数据先进行预处理, 将其转换为易于处理的形式. 另外, 用于解决分类问题的粒度智能体需要设计三方面的内容, 分别为智能体的含义、智能体的能量以及为达到目的所能采取的行为[6,7].

3.1　数据预处理

对于包含连续属性的数据集, 我们采用连续属性离散化[8]的方法对其进行离散化. 设连续属性的取值范围为$[v_{\min}, v_{\max}]$, 采样点个数为 N, 则取值为 v 的连续属性可按照式(1)离散化为 v'.

$$v' = \begin{cases} v_{\min} + \mathrm{sp} \times \mathrm{int}(v/\mathrm{sp}), & \text{if } v/\mathrm{sp} - \mathrm{int}(v/\mathrm{sp}) < 0.5 \\ v_{\min} + \mathrm{sp} \times [\mathrm{int}(v/\mathrm{sp}) + 1], & \text{if } v/\mathrm{sp} - \mathrm{int}(v/\mathrm{sp}) \geqslant 0.5 \end{cases} \quad (1)$$

其中 $\mathrm{sp} = (v_{\max} - v_{\min})/(N-1)$, 为属性取值采样间隔; $\mathrm{int}(x)$ 为小于实数 x 的最大整数.

对于包含缺失属性的数据集, 利用按概率的统计方式对其进行修补. 首先统计该属性中各个取值出现的概率, 然后以轮盘赌方式选择一个属性值, 为缺失的属性值进行填补.

3.2　粒度智能体定义

当采用粒度智能体解决分类问题时, 需要通过对粒度智能体的进化将具有最大相似特性的数据集中到一起, 然后对每类数据提取规则, 再利用这些规则对新的数据进行分类和预测. 因此, 这里的粒度智能体为数据的集合, 具体定义如下:

定义 3　粒度智能体定义为 Agent= (P, K, G), 其中 $P = \{\mathrm{data}_1, \mathrm{data}_2, \cdots, \mathrm{data}_n\}$ 为 Agent 时刻所包含的数据内容, data_i 代表一条分类数据, $1 \leqslant i \leqslant n$; $K = [\mathrm{same}, \mathrm{use}]$ 为相同属性和有用属性的集合; 目标 G 为最大化智能体的能量. 其中相同属性和有用属性的定义为:

定义 4　粒度智能体分量 P 中所包含数据取值均相同的属性即为相同属性, 其集合记为 same; 若 $c \in \mathrm{same}$ 且按照一定规则(该规则在 3.3 节中说明) c 被判为可参与粒度智能体能量的计算, 则称 c 为有用属性, 其集合记为 use.

为了达到进化分类的要求, 需要对每一类进行训练的数据形成一个粒度智能体子系统, 因此在该算法中, 是 k(k 为类别数)个智能体子系统同时协同进化的. 但是, 从另一层面上来说, 每个子系统的进化是相对独立的, 子系统的协同是为了完成属性重要度 SA 的进化. 因此, 在下文的描述中, 若无特殊标识, 均以一个子系统为例进行说明.

3.3　粒度智能体能量计算

借鉴文献[4]中对适应度函数的分析, 粒度智能体的能量与两个方面有关, 一是该智能体包含数据的条数; 二是其所包含有用属性的个数及其这些属性的重要程度. 因此对粒度智能体的能量定义如下.

定义 5　粒度智能体 Agent 的能量为:

$$\mathrm{energy}(\mathrm{Agent}) = |\mathrm{Agent}| \cdot \prod_{j=1}^{|\mathrm{use}|} \mathrm{SA}_j \quad (2)$$

其中, $|\mathrm{Agent}|$ 为该粒度智能体所包含的数据条数; use 为其中的有用属性集合; SA_j 为该集合内第 j 条属性对分类决策的重要程度.

属性重要度 SA 是随着粒度智能体的进化而不断进化的. 各个属性重要度的取值范围为[1, 5], 且在初始时均被赋值为 3, 其进化过程可描述如下:

1. 在 same 中选择一个属性 s, 并假设其取值为 x;
2. 从另一子系统中随机选取 r 个粒度智能体, 所包含的数据总条数记为 N_r, 并假设该 N_r 条数据中属性 s 取值为 x 的数据条数为 C_p;
3. 若 $C_p = 0$, 则令 $SA(s) = SA(s) * 0.8 + 1$, 属性重要度增加, 并将该属性加入到有用属性集合当中, use ← s;
4. 若 $C_p \neq 0$, 则令 $SA(s) = (1 - 0.2 \times C_p/N_r) \times SA(s) + 0.2 \times C_p/N_r$; 相应地减少该属性的重要程度.

从该过程可得知, 属性重要度在 1 到 5 之间取值, 其进化操作为超群层面上的进化, 以各子系统之间的协同来完成属性重要度的更新.

3.4 粒度智能体行为

为了达到其目的, 一个子系统中的每个粒度智能体都将与其它粒度智能体开展竞争和合作, 以便能够获得更多的资源. 同时, 粒度智能体具有自主性和进化性, 能够修改自己的形式以适应新的环境. 根据这些特性以及分类问题的要求, 我们为粒度智能体设计了 3 个进化算子以完成其进化. 其中, 同化算子实现了粒度智能体间的竞争操作; 交换算子和分化算子分别体现了粒度智能体的协作行为和自学习行为. 另外, 这些算子均直接作用于 Agent 的 P 分量上, 因此若无特别说明, 以下操作中的 Agent 即为 P 分量.

3.4.1 同化算子 **assimilation**

存活在系统中的粒度智能体, 为了获取自身的最大利益, 即包含更多具有类似特征的数据, 它将与其它粒度智能体之间发生竞争操作, 具体过程以同化算子来描述.

同化算子:
1. 选择子系统中的粒度智能体 $Agent_a$, 并依次计算该智能体与其它 Agent 中相同属性的重合程度(两者中相同且取值相等的属性越多, 则重合程度越高);
2. 按照轮盘赌的概率选择与 $Agent_a$ 发生作用的 $Agent_b$;
3. 令 $Agent_a = [Agent_a; Agent_b]$, 同时直接淘汰 $Agent_b$; 重新计算 $Agent_a$ 的相同属性及有用属性, 放入知识库 K 中.

3.4.2 交换算子 **exchange**

除了竞争之外, 两个粒度智能体可以通过互相协作以获得各自利益的增加(不一定两者同时增加), 具体过程以交换算子来描述.

交换算子:
1. 在同一子系统中随机选择两个粒度智能体 $Agent_{p1}$ 和 $Agent_{p2}$;
2. 若 $|Agent_{p1}| > m_{e1}$ 且 $|Agent_{p2}| > m_{e2}$, 则从 $Agent_{p1}$ 和 $Agent_{p2}$ 中分别挑选出 m_{e1} 和 m_{e2} 条数据进行互相交换, 得到 $Agent_{c1}$ 和 $Agent_{c2}$;
3. 令 $penergy = \max(energy(Agent_{p1}), energy(Agent_{p2}))$, $cenergy = \max(energy(Agent_{c1}), energy(Agent_{c2}))$, 若 $cenergy > penergy$, 则以 $Agent_{c1}$ 和 $Agent_{c2}$ 替代原有的 $Agent_{p1}$ 和 $Agent_{p2}$, 否则保留原有的智能体不变;
4. 计算新形成粒度智能体的相同属性和有用属性, 放入知识库 K 中.

3.4.3 分化算子 **differentiation**

除了以上提到的竞争和协作操作之外, 智能体具有自主性及其进化性, 可以通过对自身的学习改善自己, 使其更适应变化的环境, 具体体现为分化. 即当一个粒度智能体中的数据不包含任何有用属性时, 对该智能体做规则提取是毫无意义的.

分化算子:
1. 在系统中随机选择一粒度智能体 $Agent_{p1}$;
2. 如果 use = Ø 则转 3, 否则停止;
3. 从 $Agent_{p1}$ 中随机选择 m_d 条数据组成一个新的粒度智能体 $Agent_{c2}$, $Agent_{p1}$ 中除去该 m_d 条数据组成另一粒度智能体 $Agent_{c1}$;
4. 计算新形成粒度智能体的相同属性和有用属性, 放入知识库 K 中.

3.5 分类规则提取

为了提取尽可能少的规则, 同时保证所提取的规则能够最大限度地覆盖训练数据, 必须对所得到的粒度智能体进行合并. 若某一粒度智能体的相同属性包含在另一粒度智能体的相同属性中, 则将两个智能体合并成一个智能体, 具体可描述为: if $same(Agent_1) \subseteq same(Agent_2)$, 且在这些属性上的取值均相同, 则令 $Agent = Agent_1 \cup Agent_2$. 合并完之后从每个粒度智能体中提取出一条规则, 同时计算其支持度, 并按从大到小进行排序. 另外, 在不同的子系统内提取出来的分类规则有可能发生冲突, 因为从不同类中提取出来的规则所覆盖的样本有可能出现重复, 因此, 在此引入了匹配度的定义, 以便更好地预测新样本的类别.

定义 6 设样本为 d, 规则为 r, $|condition_r|$ 表示规则 r 中条件的个数, $|condition_r^d|$ 表示数据 d 所满足的条件个数, 这样规则 r 与数据 d 的匹配度为 $MV_r^d = |condition_r^d| / |condition_r|$.

匹配度的取值范围为[0, 1], 0 为匹配最差的情况, 1 为匹配最好的情况. 选取匹配度最大的规则来预测新数据的类别, 当多条规则的匹配度同样为最大时, 选取排在最前面的规则即可.

3.6 粒度智能体进化分类算法

假设数据集中共包含 k 个类别的数据, 粒度智能体进化分类算法可描述如下:

1. 初始化, 对每条数据都形成一个粒度智能体 Agent, 并将同类数据所形成的智能体放入一子系统 $MAGS_i$ 中, $i=1, \ldots, k$. 令 $t \leftarrow 0$, $i \leftarrow 1$.
2. 如果 $i > k$, 则转步骤 6;
3. 对 $MAGS_i$ 中的粒度智能体计算能量, 并按序进行同化、交换操作;
4. 对 $MAGS_i$ 中 use= Ø的智能体 Agent, 进行分化操作;
5. $i \leftarrow i+1$, 转步骤 2;
6. 判断终止条件是否满足, 如果 $t > N$, 则转步骤 7; 否则令 $t \leftarrow t+1$, $i \leftarrow 1$, 转步骤 3;
7. 从最终形成的粒度智能体中提取规则, 并对数据进行分类预测.

4 实验结果与分析

从 UCI 数据集(UC Irvine Machine Learning Repository)[9] 中选取了 13 个数据集来测试算法的预测准确性, 其参数如表 1 所示. 这些数据集包含了各种类型的分类问题, 已被广泛地用来测试分类算法的性能.

表 1 UCI 测试集

Data set	Size	Attributes			Class	Class distribution
Breast cancer	286		9*N*		2	201, 85
Vote	435	16*B*			2	267, 168
Credit	690		9*N*	6*C*	2	307, 383
Diabetes	768			8*C*	2	500, 268
Tic-tac-toe	958		9*N*		2	479, 479
Mushroom	8124		22*N*		2	4208, 3916
Iris	150		4*N*		3	50, 50, 50
Monk-1	432				3	144, 144, 144
Monk-2	432				3	144, 144, 144
Monk-3	432				3	144, 144, 144
Splice	3190		61*N*		3	767, 768, 1655
Lymph	148	9*B*	9*N*		4	2, 81, 61, 4
Zoo	101	15*B*	2*N*		7	41, 20, 5, 13, 4, 8, 10

为了准确的测试算法性能, 实验均采用 10 次交叉验证法. 对于数据集 Mushroom 和 Splice, 迭代次数 N 设为 200, 其它数据集的迭代次数均为 100 代. 实验运行在 3.2GHz, 2GB 内存的个人计算机上, 其运行环境为 MATLAB 7.4.0. 表 2 给出了对这 13 个数据集的测试结果, 包括 10 次独立运行的平均预测准确率、标准方差、平均训练时间以及所提取出的平均规则数.

表 2 GAEC 在 UCI 数据集上的测试结果

Data sets	Breast	Vote	Credit	Diabetes	Tic-tac-toe	Mushroom	Iris	Monk-1	Monk-2	Monk-3	Splice				Lymph	Zoo
											EI	IE	NE	All		
Predictive accuracy(%)	98.25	97.04	90.24	86.42	100.00	100.00	100.00	100.00	79.28	100.00	96.73	96.35	96.47	96.44	91.25	92.15
Standard deviation(%)	2.00	1.96	2.36	3.04	0.00	0.00	0.00	0.00	3.22	0.00	1.43	0.79	1.02	1.11	7.25	1.05
Training times(s)	1.12	0.21	1.44	0.52	0.43	12.11	0.17	0.21	0.72	0.17	101.21				0.18	0.07
Number of rules	14.7	4.3	16.9	4.9	10.1	14.5	7.8	7.8	32.4	5.1	41.7				9.5	11.1

分析表 2 中的测试结果, 在测试的 13 个 UCI 数据集中, 可对其中的 5 个数据集(Monk-1, Monk-3, Tictactoe, Mushroom 和 Iris) 达到 100% 的分类预测率; 对于其余暂时未能完全准确分类的数据集, 其平均预测准确率大多都在 90% 以上. 对于带类别噪声的数据集 Monk-2, 其平均预测准确率较低, 在 80% 左右. 究其原因是算法总体指导思想仍是将相似的数据聚集, 然后从中提取规则, 所以当其中含有类别噪声时, 提取的规则可能会存在一定的偏颇. 另外, 从表中可以看到, 各次实验所得的预测准确率的标准方差较小, 说明该算法具有较强的稳定性. 对于大部分数据集, 其训练时间代价很小; 对较大规模的数据集 Mushroom 以及 Splice, 训练时间也仅为 12.1s 和 101.21s. 从所提取的平均分类规则数目来看, 其中有六个数据集的分类规则数在 10 以下; 除了 Monk-2 和 Splice 之外, 其它 5 个数据集的分类规则条数均为十几条. 总之, 以较少的规则数即可达到较高的分类预测准确率, 这样的结果是比较令人满意的.

表 3 GAEC 与其它算法(C4.5, G-NET, OCEC) 的比较结果

Data sets		Predictive accuracy±Standard deviation			
		C4.5	G-NET	OCEC	GAEC
Breast cancer		94.15±3.32	94.71±2.89	96.13±2.03	98.25±2.00
Vote		95.37±3.05	94.90±3.20	95.87±2.61	97.04±1.96
Credit		85.97±3.28	84.20±4.40	87.97±4.38	90.24±2.36
Diabetes		79.8±1.7	–	–	86.42±3.04
Tic-tac-toe		92.93±1.82	99.03±0.62	100.00±0.00	100.00±0.00
Mushroom		100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Iris		100.00±0.00	–	–	100.00±0.00
Monk-1		100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Monk-2		67.17±10.66	97.20±3.80	73.18±7.31	79.28±3.22
Monk-3		100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Splice	EI	–	96.60	95.98	96.73
	IE	–	97.10	94.98	96.35
	NE	–	96.70	95.67	96.47
	All	93.8	–	93.34±0.52	96.44±1.11
Lymph		79.8±8.4	–	86.38±8.92	91.25±7.25
Zoo		90.9±8.4	–	–	92.15±1.05

在已有的分类算法中,G-NET[10]是一种性能良好的基于进化的分类方法; OCEC[4]是一种自下而上的协同分类算法,比一般的进化分类算法更具优越性; 除此之外,C4.5[11]是一种性能良好,且比较稳定的算法,已经在各个领域内广泛使用,因此,实验中同时将本文算法与这三种算法进行比较,分别采用平均预测准确率和标准方差两个指标来评价,结果如表 3 所示.

从表3的结果中可以看出,对于大部分的测试数据集,GAEC 的性能在进行比较的 4 个算法中都达到了最优. 对于其中的Monk-1, Monk-3, Tictactoe, Mushroom 以及 Iris 这五个数据集的识别率都达到了 100%; 和广泛使用的 C4.5 相比,这 13 个数据集的识别率都与其相当或者要更高一些; 和 G-NET 相比的 9 个数据集中,除了 Monk-2 和 Splice 外,GAEC 对其余 7 个数据集的平均识别率都要高于 G-NET; 和 OCEC 相比较的 10 个数据集中,GAEC 的识别率更高一些. 尽管 GAEC 和 OCEC 这两者均采用了自下而上的分类方式,且都遵循将具有最大相似性的数据集中在一起,然后从中提取规则的整体思想,但OCEC 中,对于"组织"的选择和操作都是随机的; 而 GAEC 中利用了"粒度智能体"作为操作的基本单位,在其中增加了知识库,整个进化过程是随机性和指导性并存的,有利于相似数据的快速聚类. 同时,知识库的引导使得数据的聚集更具方向性. 除此之外,从表中可以看出,除开数据集 Lymph,GAEC 对其它数据集预测率的标准方差均在 3% 左右或以下,说明在 10 次交叉验证中,算法具有较强的稳定性.

5 算法分析

在粒度智能体进化分类算法中,设计了三种粒度进化算子来完成整个进化过程. 为了进一步明确三种算子在整个进化过程中所起的作用,下面对三种算子的性能作进一步分析. 首先是同化(竞争)算子,初始化时我们将每条数据作为一个粒度智能体加入到智能体系统当中,如果没有该算子,整个进化过程就无法进行,因此该算子的重要性是显而易见的. 从智能体系统方面来解释,智能体间正是通过这样子的竞争同化(竞争)算子将智能体的局部信息进行传递,从而扩散到整个智能体系统当中,而这一点也十分符合自然进化模型. 其次是交换(协同)算子,该算子使得粒度智能体通过吸收环境中其它粒度智能体中的某些有用信息来提升自身的能量,符合整个自然系统中协同竞争的思想. 除了这两个算子之外,还有一个很重要的算子——分化(自学习)算子. 当某个粒度智能体中不包含有用属性时,将执行此操作. 换言之,根据本文所定义的规则提取方法,这样的粒度智能体中无法提取出有用的规则,因此必须将其中的数据分散,并以各种形式加入到别的粒度智能体中,才能达到使相似性数据聚集的目的. 从以上分析来看,这三个进化算子从不同的方面一同协助完成整个进化过程,起到了不同程度的作用,但这三者是相辅相成,缺一不可的. 另外,这三个算子均为简单的合并分裂等操作,因此算法总的复杂度为 $O(N)$.

另一方面,本文和 OCEC 同样采用了自底向上的分类思想,即通过将相似的数据聚类,然后从其中提取规则,并对新数据进行预测分类. 但在完成数据聚类的进化过程中,两者存在着一些差别. 首先,本文的进化算子来自于文化进化的指导,同时完成了两个层面上的进化,群进化(同一类别数据的单独进化)和超群进化(各粒度智能体子系统之间协同,完成属性重要度进化,用以支持下一步的群进化). 而 OCEC 中的组织思想来源于经济学,整个进化操作仍源于遗传操作的指导; 其次,具体的进化算子不同,GAEC 设计了三个操作算子,通过一种有序的操作完成进化过程,且算子的操作融合了知识库 *KD* 的指导作用,能够更快速地将具有相似属性的数据聚集到同一粒度智能体中. 而 OCEC 中采用了完全随机的操作,随机选择发生作用的智能体,随机选择所需的操作算子,这种做法虽然在一定程度上可以完成相似数据聚类的目的,但势必会增加规则训练的时间以及不确定性; 再者,属性重要度的进化过程稍有不同. 两者的进化思想大致相同,但 GAEC 中的进化方式相对简单,能快速完成属性重要度进化的目的.

6 小结

受粒度进化计算的启发,同时引入智能体的概念,为数据挖掘中的分类任务提出了一种粒度智能体进化分类方法. 该方法通过粒度智能体的进化将具有最大相似度的数据聚集到一起,同时采用知识库的方式来指导进化. 根据粒度进化的要求以及智能体的工作特点,设计了三个算子——同化算子、交换算子以及分化算子来协同完成整个进化过程,这三个算子分别体现了智能体间的竞争行为、协作行为以及自学习的能力. 在完成粒度智能体的进化之后,通过一定策略从中提取出规则用以对新数据的预测分类. 对 UCI 数据集的测试结果表明该算法能以较小的代价获得较有效的分类规则.

参考文献:

[1] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques[M]. San Mateo, USA: Morgan Kaufmann Publishers, 2000. 185- 211.

[2] Linyu Yang, Widyantoro, D H, Loerger T, Yen J. An entropy-based adaptive genetic algorithm for learning classification rules

[A]. Proceedings of IEEE Congress on Evolutionary Computation[C]. Seoul, South Korea, 2001. 790– 796.

[3] Deborah R. Carvalho, Alex A. Faraeitas. A hybrid decision tree/ genetic algorithm method for data mining[J]. Information Sciences, 2004. 163(1): 13– 35.

[4] 刘静, 钟伟才, 刘芳, 焦李成. 组织协同进化分类算法[J]. 计算机学报, 2003, 26(4): 446– 453.
LIU Jing, ZHONG Wei-Cai, LIU Fang, JIAO Li-Cheng. Classification based on organizational coevolutionary algorithm[J]. Chinese Journal of Computers, 2003, 26(4): 446– 453. (in Chinese)

[5] 蒙祖强, 蔡自兴. 一种新的计算方法: 粒度进化计算[J]. 计算机工程与应用, 2006, 42(1): 5– 8.
Meng Zuqiang, Cai Zixing. A new computing: Granular evolutionary computing[J]. Computer Engineering and Applications, 2006, 42(1): 5– 8. (in Chinese)

[6] Liu Jiming, Han Jing, Tang Yuan Y. Multi-agent oriented constraint satisfaction[J]. Artificial Intelligence, 2002, 136(1): 101 – 144.

[7] Weicai Zhong, Jing Liu, Mingzhi Xue, Licheng Jiao. A multiagent genetic algorithm for global numerical optimization[J]. IEEE Transactions on Systems, Man and Cybernetics, 2004, 34(2): 1128– 1141.

[8] J Y Ching, A K C Wong, K C C Chan. Class-dependent discretization for inductive learning from continuous and mixed-mode data[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(7): 641– 651.

[9] UCI repository of machine learning databases[DB/OL]. http://www.ics.uci.edu/~ mlearn/MLRepository.html 2007-07-14.

[10] Angalano C, Giordana A, Lo Bello G, Saitta L. An experimental evaluation for coevolution concept learning[A]. ICML'98: Proceedings of the Fifteenth International Conference on Machine Learning[C]. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1998. 19– 27. (in Chinese)

[11] Quinlan J R. C4.5: Programs for Machine Learning[M]. San Mateo, CA: Morgan Kaufmann, 1993.

作者简介:

潘晓英　女, 1981 年 10 月出生于浙江丽水. 现为西安电子科技大学博士研究生. 主要研究方向为多智能体系统, 自然计算, 数据挖掘等.
Email: xiaoyingpan@gmail.com

焦李成　男, 1959 年 10 月出生于陕西白水. 西安电子科技大学教授, 博士生导师, IEEE 高级会员. 主要研究方向为智能算法, 机器学习, 非线性科学, 智能信号处理, 小波理论及其应用.
Email: lchjiao@mail.xidian.edu.cn

刘　芳　女, 1963 年 2 月出生于湖南华容. 现为西安电子科技大学教授, 博士生导师, 主要研究方向为人工智能, 信号和图像处理, 模式识别和进化计算.
Email: f63liu@163.com