文章编号: 1000-6788(2012)10-2215-06 中图分类号: O231.1 文献标志码: A

采用并行协同进化遗传算法的文本特征选择

邬开俊 [1], 鲁怀伟 [2]

(1. 兰州交通大学 电子与信息工程学院, 兰州 730070; 2. 兰州交通大学 数理与软件工程学院, 兰州 730070)

摘 要 现有的文本特征选择方法都是串行化的, 应用于海量文本数据集时时间效率较低, 因此利用并行思想来提高文本特征选择的效率, 已成为文本挖掘领域的一个研究热点. 本文将遗传算法和并行协同进化算法结合起来, 在粗糙集的基础上设计了一个并行协同进化遗传算法并将该算法用于文本特征选择. 该方法采用遗传算法搜索特征, 利用并行协同进化算法来提高时间效率, 从而较快地获得较具代表性的特征子集. 实验结果表明该方法是有效的.

关键词 特征选择; 文本挖掘; 遗传算法; 协同进化; 粗糙集

PCEGA used to solve text feature selection

WU Kai-jun[1], LU Huai-wei[2]

(1. School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China;
2. School of Physics and Software Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract Most of existing text feature selection methods are serial and are inefficient timely to be applied to Chinese massive text data sets. So, it is a hotspot of text mining how to improve efficiency of text feature selection by means of parallel thinking. Combining genetic algorithm with parallel collaborative evolutionary, a parallel collaborative evolutionary genetic algorithm (PCEGA) based on rough sets was designed and used to select text features. The presented method took advantage of genetic algorithm to select features and employed parallel collaborative evolutionary to enhance time efficiency, so that the more representative feature subsets was acquired quickly. Experimental results show that the method is effective.

Keywords feature selection; text ming; genetic algorithm; collaborative evolutionary; rough sets

1 引言

随着计算机的普及和互联网的发展, 网络上积累了越来越多的海量数据, 这些海量数据绝大多数是以文本形式存在的. 如何从这些海量文本数据中寻找到潜在有价值的信息, 引起了国内外专家学者的关注. 在这种情况下, 文本挖掘应运而生并成为了一个研究热点 [1]. 在文本挖掘中, 文本分类是被应用较为广泛的一个技术. 文本分类是自动地为待分类的文本分配一个或多个类别的过程, 这个过程中待分类的文本常用空间向量模型来表示 [2]. 由于文本转化成空间向量后其维数通常巨大而且存在大量空值, 这必然造成文本分类计算开销的剧增、效率的低下 [3]. 为此, 寻找一种高效的特征选择方法, 以降低特征空间维数、避免维数灾难, 提高文本分类的效率和精度, 成为文本分类中亟待解决的重要问题.

现存诸多特征选择方法都是串行化的, 例如互信息、信息增益等 [4], 遇到海量文本数据集时效率较低, 所以, 研究并行化特征选择方法十分有必要.

从海量文本集中选择较具代表性特征子集的问题是一个最优化问题, 对于这类问题, 遗传算法往往能够有效地加以解决 [5]. 但是, 遗传算法在求解最优化问题时并没有考虑各种群之间的相互协作, 而实际情况却是种群之间的相互协作在物种演化过程中占据着重要的地位. 模仿生物种群之间的相互协作来求解最优化问题是协同进化算法的主要思想 [6]. 本文把协同进化的思想同遗传算法结合起来, 在粗糙集的基础上提出了

收稿日期: 2010-06-25
资助项目: 国家自然科学基金 (12CGL004); 兰州交通大学青年科学研究基金 (2011005)
作者简介: 邬开俊 (1978–), 男, 山东莒南人, 副教授, 博士研究生, 研究领域: 启发式优化算法、计算机仿真; 鲁怀伟 (1959–), 男, 甘肃天水人, 教授, 博士生导师, 研究领域: 光纤通信, 光纤无源器件.

一种适用于海量文本特征选择的并行协同进化遗传算法. 实验证明了所提算法能够较快地选择出优秀的特征子集.

2 粗糙集基本理论

粗糙集理论是 20 世纪 80 年代初由 Pawlak 提出来的一种新的处理不精确、不相容、不完全和不确定知识的软计算工具, 目前它已被广泛应用于机器学习、决策分析、数据挖掘、过程控制、智能信息处理等领域 [7]. 本文把粗糙集引入并行协同进化遗传算法之中用于设计相应的适应度函数、交叉算子、变异算子. 因篇幅限制, 下面仅介绍与论文内容紧密相关的粗糙集知识, 具体请参阅文献 [7].

定义 1[7] 已知信息系统 $S =< U, C \cup D, V, f >$, 若 $U/C = \{X_1, X_2, \cdots, X_n\}$, $U/D = \{Y_1, Y_2, \cdots, Y_m\}$, 则称 D 关于 C 的依赖度为

$$\gamma_C(D) = \frac{1}{|U|} \sum_{i=1}^{m} |Pos_C(Y_i)| \tag{1}$$

其中, $Y_i \in U/D$.

定理 1[7] 已知信息系统 $S =< U, C \cup D, V, f >$, $B \subseteq C$, 若 $\gamma_B(D) = \gamma_C(D)$, 则称 B 是 C 的约简集.

定义 2[7] 属性 a 加入 $R \subseteq C$, 对于分类 U/D 的重要度定义为:

$$SGF(aRD) = \gamma_{R+\{a\}}(D) - \gamma_R(D) \tag{2}$$

$SGF(aRD)$ 越大, 表明在条件属性集 R 下, 属性 a 对决策类 D 就越重要.

3 协同进化简介

自然界中各物种之间存在着彼此制约、彼此促进的关系, 这种关系称为协同进化 [8]. Hillis 于 1990 年将协同进化引入到搜索机制中, 提出了协同进化遗传算法 (CGA). 在 CGA 中, 有多个协同进化的种群个体, 不同物种的个体间既有共同合作进化又存在相互竞争. 后来, Paredis 又在 CGA 中采用了生命周期适应度评价方法, 这使得 CGA 的健壮性更强并在众多问题中得到了成功应用, 如分类问题、约束问题等 [9].

在最优化问题中, 通常由多个部分组成最优解或近似最优解, 每个部分称为部分解. 一个部分解表达了问题的一个方面, 该部分解只有同其它部分解合作, 才能组成最优解也即完全解. 如果在遗传算法中只进化部分解形成种群, 通过组合部分解来搜索完全解的空间, 在解决问题时就能有效地控制算法的搜索空间 [10], 此时的遗传算法就成为了协同进化遗传算法. 在该算法中, 首先将完全解分解成若干个长度相同的部分解以形成初始部分解种群, 然后执行以下进化过程: 随机从种群中抽取若干部分解放在完全解的各个位置来形成完全解; 构造好多个完全解后, 再评价每个完全解, 之后评价它的各个部分解; 接着对部分解种群进行选择、交叉、变异操作, 随后再进行下一轮进化.

在完全解的不同位置上, 部分解的最优模式通常是不同的, 这就使得部分解种群在进化过程中能保持多样性, 从而有效地防止陷入局部最优, 这是协同进化算法的优点. 该算法的不足在于: 它忽略了部分解的位置信息, 可能会将一个位置上的最优模式安排到其它位置上, 从而降低了完全解的适应度, 进而影响该最优模式的生存. 本文既考虑部分解的位置信息又兼顾部分解种群的多样性, 提出了并行协同进化遗传算法.

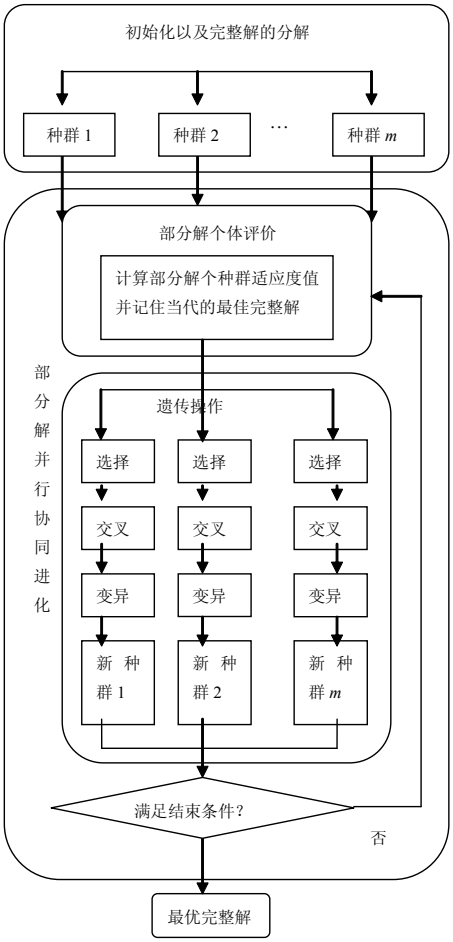

图 1 算法流程图

4 并行协同进化遗传算法

4.1 算法流程

图 1 所示的算法流程表明种群与种群之间在进化过程中的协调关系, 不但考虑了个体之间的竞争, 还考虑了个体之间的协作, 更加符合自然界进化的规律.

4.2 算法中的策略

4.2.1 编码策略

对于一个特征来说, 它要么出现在被选择的特征集中, 要么就不出现, 仅这两种选择, 所以对染色体采用 0/1 编码比较合理. 具体编码如下: 已知原始特征集 $C=\{c_1,c_2,\cdots,c_n\}$, 可把每个特征的下标当做该特征的唯一编号, 此时染色体是一个长度为 n 的 0/1 字符串, 在相应的编号中如果是 1, 则该编号对应的特征就被选用, 否则就被弃用. 这样, 每条染色体就同一个特征子集相对应.

4.2.2 部分解形成策略

假设种群规模为 m, 因每个种群对应一个完整解, 则有 m 个完整解. 由于初始特征集有 n 个特征组成, 则每个部分解种群的长度可为 $L=n/m$. 若 $n\%m$ 不为零, 则把这 $n\%m$ 个特征分配给前 $n\%m$ 个种群, 每个部分解种群分一个, 这样在并行计算时就考虑了负载平衡问题. 对每个部分解种群随机选择一个随机数 P, 产生 P 个长度为 n/m 或 $(n/m)+1$ 的 0/1 字符串, 作为初始种群.

4.2.3 完整解适应度函数设计策略

在特征选择中, 如果一个特征子集的元素个数越少并且类别集对该特征子集的依赖度越大, 该特征子集代表性就越好, 它对应的完整解就越优秀, 此时完整解的适应度值就应该越大, 所以定义如下适应度函数:

$$F(x)=\begin{cases}\dfrac{Card(C-B(x))}{Card(C)}\gamma_{B(x)}(D), & \text{若 } \gamma_C(D)-\gamma_{B(x)}(D)\geq\varepsilon_1\\ \dfrac{2\times Card(C-B(x))}{Card(C)}\gamma_{B(x)}(D), & \text{若 } \gamma_C(D)-\gamma_{B(x)}(D)<\varepsilon_1\end{cases}\tag{3}$$

其中, ε_1 表示精度误差, 文中令 $\varepsilon_1=0.01$; $B(x)$ 表示个体 x 中对应位为 1 的特征组成的集合; $\gamma_{B(x)}(D)$ 表示类别集 D 关于特征集合 $B(x)$ 的依赖度. 特征集 $B(x)$ 元素的数越少、类别集 D 关于特征集 $B(x)$ 的支持度越大, 适应度函数值也就越大, 此时适应度函数中的 $\frac{Card(C-B(x))}{Card(C)}\gamma_{B(x)}(D)$ 部分正好体现了这一思想. 若 $\gamma_C(D)-\gamma_{B(x)}(D)<\varepsilon_1$ 则表明所选特征集 $B(x)$ 较为接近最优解, 此时应该适当提高完整解 x 的适应度以便使它朝着更有利的方向进化.

4.2.4 部分解适应度函数设计策略

已知种群 a, 则属于该种群的部分解 b 的适应值计算策略如下: 首先从除种群 a 以外的其它种群中各随机选出一个部分解同部分解 b 组成问题的完整解, 然后按照公式 (3) 计算这个完整解的适应度; 将这个过程循环执行 L 次后, 取所获得的 L 个完整解适应度平均值作为部分解 b 的适应度.

4.2.5 选择算子设计策略

采用轮盘赌方式实现选择算子. 如果染色体规模为 m, 用 $G(x)=\{x_1,x_2,\cdots,x_n\}$ 表示个体 x_j 的适应度为 $F(x_j)$, 那么它被选择的概率为:

$$P(x_j)=\frac{F(x_j)}{\sum_{i=1}^{m}F(x_j)},\quad j=1,2,\cdots,m\tag{4}$$

使用选择操作生成交配池以用于繁殖下一代, 其中父代个体在下一代的生存期望数目为: $G(x_j)=m\times P(x_j), j=1,2,\cdots,m$. 选择过程体现了 “优胜劣汰, 适者生存” 的自然进化法则, 保证了优秀基因能够传递给下一代. 而且, 由于在进化过程中采用了轮盘赌选择策略, 从而使得个体的适应度越大, 其被选择的概率也就越大, 这有利于算法快速收敛到最优解.

4.2.6 自适应交叉算子设计策略

交叉操作可以使父代的良好基因片段信息得以保存, 从而繁殖出更好的子代. 由于特征集中的每个特征的重要程度有所不同, 本文就选择特征重要度作为设计交叉算子的基础. 已知个体: $x_i=\{x_{i1},x_{i2},\cdots,x_{in}\}$, $x_j=\{x_{j1},x_{j2},\cdots,x_{jn}\}$, 则交叉操作如下:

$$O(P_{ij},s_{ij}):\begin{cases} x_{ik}=\begin{cases} x_{ik}, & s_{ij}\geq P_{ij},\\ x_{jk}, & s_{ij}<P_{ij}, \end{cases}\\ x_{jk}=\begin{cases} x_{jk}, & s_{ij}\geq P_{ij},\\ x_{ik}, & s_{ij}<P_{ij}, \end{cases} \end{cases} \quad k=1,2,\cdots,n \tag{5}$$

其中 P_{ij} 为染色体 x_i 和染色体 x_j 交叉操作时的交叉概率, 这里令: $P_{ij}=(\gamma_{B(x_i)}(D)+\gamma_{B(x_j)}(D))/2$; $s_{ij}\in[0,1]$ 代表染色体 x_i 和染色体 x_j 交叉操作时的均匀随机概率变量.

4.2.7 自适应变异算子设计策略

在遗传算法中, 通过按变异概率 P_m 随机反转某位的二进制字符值来变异操作, 对于给定的染色体 $x_i=\{x_{i1},x_{i2},\cdots,x_{in}\}$, 具体变异操作如下:

$$O(P_{ik},s_{ik}):x_{ik}\begin{cases} x_{ik}, & s_{ik}\geq P_{ik},\\ 1-x_{jk}, & s_{ik}<P_{ik}, \end{cases} \quad k=1,2,\cdots,n \tag{6}$$

其中 P_{ik} 表示染色体 x_i 第 k 位变异操作时的变异概率, 这里令 $P_{ik}=1-SGF(B(x_{ik}),B(x_i),D)$, $B(x_{ik})$ 为染色体 x_i 第 k 位对应的特征, 如果第 k 位为 0, 则 $B(x_{ik})=\emptyset$, 否则 $B(x_{ik})$ 就为第 k 位对应的特征; $s_{ik}\in[0,1]$ 是染色体 x_i 第 k 位变异操作时的均匀随机概率变量.

4.2.8 中止条件

以预先设定的最大繁殖代数作为停止准则.

4.3 算法过程描述

Step 1 初始化: 最优完全解 BestSolution 的适应度 BFitness=0, m 个初始种群, 每个种群的规模为 40, 最大迭代次数 MAX1, 允许 BestSolution 连续没变的次数 MAX2, 当前迭代次数 $i=0$, BestSolution 当前没改变的次数 $j=0$.

Step 2 按照 4.2.4 节的方案计算各种群中各个个体的适应值.

Step 3 从每个种群随机选出一个个体组成完全解, 按照 4.2.3 节的方案计算该完全解的适应值, 如果该适应值大于 BFitness, 则令 BFitness 等于该值, 用 BestSolution 记住该完全解, $j=0$; 否则 $j=j+1$. 如果 $j\geq$MAX2, 则转 Step 6, 否则, 该过程继续重复, 直到达到 50 次.

Step 4 对各个种群进行选择、交叉、变异这三种操作, 它们分别使用公式 (4)、(5)、(6), 形成新种群.

Step 5 $i=i+1$, 如果 $i\geq$MAX1, 则转 Step 6; 否则转 Step 2.

Step 6 输出 BestSolution 对应的特征集, 算法结束.

5 实验例证

5.1 实验语料库

采用复旦大学计算机信息与技术系国际数据库中心自然语言处理小组构建的中文文本分类语料库作为实验数据, 其免费下载网址为: http://www.nlp.org.cn/categories /default.php?cat_id=16. 该语料库由 20 个类别的 14378 篇文档组成, 其中 6164 篇为测试文本集, 8214 篇为训练文本集; 各类别的测试文本集和训练文本集之间互不重叠, 也即一篇文档仅属一个文本集并且每篇文本仅属于一个类别. 这个语料库各类别训练文档数相差非常大, 其中训练文档数较小的类别占大多数, 约为 11 个类别, 它们的训练文档数均少于 100 篇, 例如通信类训练文档数仅有 25 篇.

5.2 实验环境及参数设置

中文分词处理采用的是中科院计算所开源项目汉语词法分析系统 ICTCLAS 系统. 实验使用的软件工具是 Weka, 这是纽西兰的 Waikato 大学开发的与数据挖掘相关的一系列机器学习算法, 它是开源项目, 网址为: http:// www.cs.waikato.ac.nz/ml/weka/. 实验使用的计算工具为 Matlab 7.0. 经试验, 算法中各参数设置如下: $\varepsilon_1=0.01$, 种群数为 40, 每个种群的规模为 80, 最大迭代次数 MAX1= 100, 允许 BestSolution 连续没变的次数 MAX2= 20. 文中采用的并行计算方法为文献 [11] 中的主从式方法.

5.3 实验所用分类器及其评价标准

实验中, 使用信息增益 (IG)、x^2 统计量 (CHI)、互信息 (MI) 这三种经典的特征选择方法与本文方法进行优劣比较. 因 KNN 分类器简单、易理解, 本实验就在各种特征选择方法后采用 KNN 分类器 (K 设置为

20, 采用两向量的夹角作为文本间的距离) 来对这四种特征选择方法进行比较.

实验中采用宏平均 F_1 和微平均 F_1 作为分类器性能评价标准. 在时间性能方面, 串行算法采用算法整个执行过程所消耗的时间, 并行算法采用加速比 S_p 与效率 E_p 分析: 加速比是评价算法的并行性对运行时间改进的程度, 用公式表示为 $S_p = T_s/T_p$, 其中 T_p 是求解一个问题的并行算法在最坏情况下的运行时间, T_s 是求解同一个问题的串行算法在最坏情况下的运行时间; 效率反映了并行系统中处理器的利用情况, 用公式表示为 $E_p = S_p/p$, 其中 p 为处理器的个数.

5.4 实验结果及其分析

5.4.1 分类器性能仿真对比结果及其分析

从各特征选择方法选出的特征集 (特征已按权重倒序排列) 中分别选择不同数目的特征对实验数据进行宏平均 F_1 和微平均 F_1 计算, 结果如图 2 和图 3 所示.

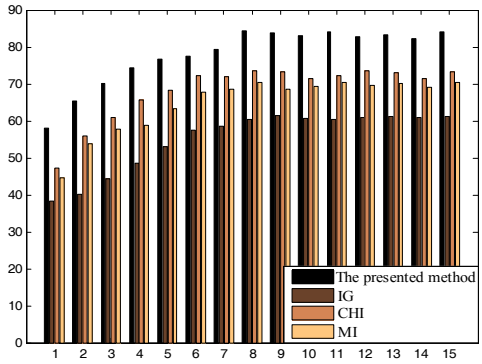

图 2　宏平均 F_1 对比结果

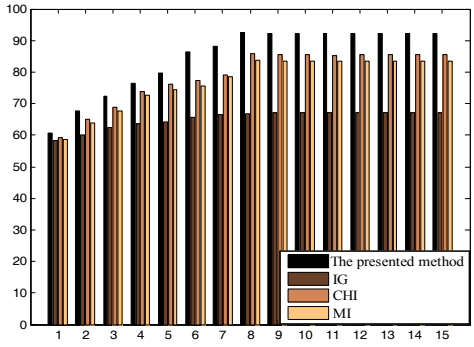

图 3　微平均 F_1 对比结果

其中两图纵轴分别代表宏平均 F_1 值和微平均 F_1 值 (单位: %), 两图横轴相应的数字代表的特征数为: (1)50、(2)100、(3)200、(4)500、(5)800、(6)900、(7)1000、(8)1500、(9)2000、(10)2500、(11)3000、(12)3500、(13)4000、(14)4500、(15)5000.

从图 2 和图 3 可以看出, KNN 分类器性能情况如下: 在本文方法所选的前 1500 个特征上性能最佳, 宏平均 F_1 值约为 85%, 微平均 F_1 值约为 93%; 在 CHI 方法所选的前 1500 个特征上性能最佳, 宏平均 F_1 值约为 74%, 微平均 F_1 值约为 86%; 在 MI 方法所选的前 1500 个特征上性能最佳, 宏平均 F_1 值约为 70%, 微平均 F_1 值约为 84%; 在 IG 方法所选的前 2000 个特征上性能最佳, 宏平均 F_1 值约为 61%, 微平均 F_1 值约为 67%. 从这可以看出这四个特征选择方法的优劣依次为本文方法、CHI、MI、IG. 原因在于: 本文方法在选择特征时, 使用了并行协同进化遗传算法考查了特征之间的隐含关系从而有效地消除了冗余特征, 这使得本文方法受类别分布影响较小, 因此所选特征集较具代表性. CHI 方法在选择特征时不但考查了特征在文档中存在的情况而且还考查了特征不在文档中的情况, MI 方法仅考查了特征在文档中存在的情况, 但它们都没能有效地消除冗余特征. 因此, 这两个方法要劣于本文方法, 但是 CHI 方法要优于 MI 方法; 由于实验中所用语料库中各类别样本分布相差较大, 而 IG 方法对类别样本分布极其敏感, 因此, 在此情况下 IG 方法所选择的特征集代表性最差.

5.4.2 消耗时间结果对比和分析

图 4 和表 1 综合表明, 本文方法串行执行时消耗的时间高于其他三种方法, 但采用并行策略后消耗的时间要远远低于其他三种方法. 表 1 表明, 随着 CPU 个数的增加, 本文方法所消耗的时间大幅减少. 当每个 CPU 被分配给相同数目的种群时, CPU 的效率比较高, 大于 92%; 当每个 CPU 被分配给不同数目的种群时, CPU 效率呈下降趋势, 例如: 当从 CPU 个数为 16 时, 处理器效率降到 90% 左右; 当从 CPU 个数为 32 时, 处理器效率降到 84% 左右. 从整体上来说, 随着从 CPU 个数的增加, 处理器效率是下降的, 原因在于: 当相同数目的种群被分给各从 CPU 时, 这使得各从 CPU 可以较好地并行协同工作, 主 CPU 工作时务须单独等待某个从 CPU; 当不同数目的种群被分给各从 CPU, 这使得它们各自完成工作的时间也不一致, 此时系统负载不平衡, 这造成主 CPU 必须等待各从 CPU 都结束工作后才能工作. 随着从 CPU 个数的递增, 分配给每个从 CPU 的种群数也在不断减小, 使得各从 CPU 传送数据的时间在它整个执行时间中的比例增大, 从而导致各从 CPU 效率逐渐降低. 从上述分析可知, 在并行算法中, 如果仅仅为了追求时间效率而盲目增加从

CPU 个数, 那么处理器的效率就会降低, 进而造成资源的极大浪费. 因此, 在使用并行算法时, 在加速比和效率之间进行权衡是必要的.

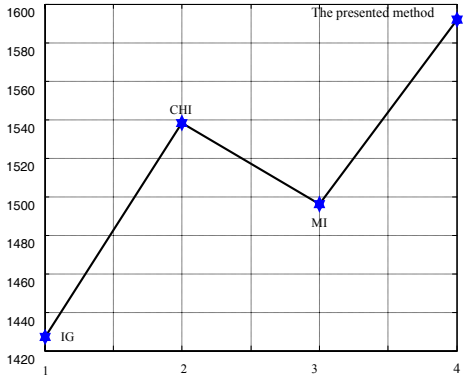

图 4 串行算法消耗的时间（单位: s)

表 1 不同处理器个数下消耗的时间、加速比、并行效率

CPU 个数	时间/s	加速比	效率
1	1592	1.00	1.0000
2	803	1.98	0.9913
4	412	3.86	0.9660
8	209	7.62	0.9522
10	172	9.26	0.9256
16	110	14.47	0.9045
20	85	19.73	0.9365
32	59	26.98	0.8432

6 结束语

本文把并行协同进化的思想同遗传算法结合起来, 在粗糙集的基础上提出了一种适用于海量文本特征选择的并行协同进化遗传算法. 该算法不但考虑了局部解的位置信息而且还兼顾了局部种群的多样性, 克服了简单遗传算法的缺点, 从而能有效地处理中文海量文本数据集, 较快地获得较具代表性的特征子集. 实验证明: 在宏平均 F_1 和微平均 F_1 方面, 本文提出的方法优于 IG、CHI、MI, 这说明本文的方法能够获得较优的特征子集; 在时间方面, 本文方法远远低于其他三种作比较的方法, 这说明它能较快地获得特征子集. 上述综合说明了本文方法具有较好的实用价值, 为文本特征选择提供一种思路.

参考文献

[1] Nguyen M H, Torre F D. Optimal feature selection for support vector machines[J]. Pattern Recognition, 2010, 43(3): 584–591.

[2] Liu H W, Sun J G, Liu L. Feature selection with dynamic mutual information[J]. Pattern Recognition, 2009, 42(7): 1330–1339.

[3] Destrero A, Mosci S, Mol C D. Feature selection for high-dimensional data[J]. Computational Management Science, 2009, 6(1): 25–40.

[4] Xu Y. A formal study of feature selection in text categorization[J]. Journal of Communication and Computer, 2009, 6(4): 32–41.

[5] Nandi B, Barman S, Paul S. Genetic algorithm based optimization of clustering in ad-hoc networks[J]. International Journal of Computer Science and Information Security, 2010, 7(1): 165–169.

[6] Lung R I, Chira C, Dumitrescu D. An agent-based collaborative evolutionary model for multimodal optimization[C]// Proceedings of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation, USA: Atlanta, 2008: 1969–1976.

[7] 胡寿松, 何亚群. 粗糙决策理论与应用 [M]. 北京: 北京航空航天大学出版社, 2006.
Hu S S, He Y Q. Rough Decision Theory and Application[M]. Beijing: Beihang University Press, 2006.

[8] Lung R I, Dumitrescu D. A new collaborative evolutionary-swarm optimization technique[C]// Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation, England: London, 2007: 2817–2820.

[9] Gog A, Dumitrescu D, Hirsbrunner B. Collaborative evolutionary algorithms for combinatorial optimization[C]// Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, England: London, 2007: 1511–1517.

[10] 于晓义, 孙树栋, 褚崴. 基于并行协同进化遗传算法的多协作车间计划调度 [J]. 计算机集成制造系统, 2008, 14(5): 991–1000.
Yu X Y, Sun S D, Chu W. Parallel collaborative evolutionary genetic algorithm for multi-workshop planning and scheduling problems[J]. Computer Integrated Manufacturing Systems, 2008, 14(5): 991–1000.

[11] 谷建军. 粗糙集理论在数据约简中的应用研究 [D]. 济南: 山东师范大学, 2007.
Gu J J. Application of rough set theory in data reduction[D]. Jinan: Shandong Normal University, 2007.