基于粗糙集和蚁群算法的特征基因选择方法

黄丹凤,祁云嵩,许姗娜
(江苏科技大学 计算机科学与工程学院,江苏 镇江 212003)

摘 要:特征基因选择在微阵列数据分析中占据着非常重要的作用,好的特征选择方法是提高基因表达数据的分类精度与分类速度的关键之一。联系蚁群算法和粗糙集理论在微阵列数据处理上的优势,文中结合粗糙集理论,对蚁群优化算法模型进行了改进,并将粗糙集的属性依赖度和属性重要度应用到蚁群算法的路径选择及评估中,提出一种新的基因选择方法。该方法实现简单,并可以比较快速地获得最优解,最终选择出较小的并且分类性能较强的特征基因子集。通过对基因数据集的仿真实验表明,该算法是有效可行的。

关键词:特征选择;粗糙集;蚁群算法
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2012)06-0068-03

Gene Selection Method Based on Rough Sets and Ant Colony Algorithm

HUANG Dan-feng, QI Yun-song, XU Shan-na
(School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: Gene selection takes an important place in DNA date analysis. Good feature selection method is one of the keys to improve the gene expression data classification accuracy and speed. Take advantage of ant colony optimization (ACO) algorithm and rough set theory in microarray data processing, use rough set theory and ACO algorithm to put forward a gene selection method. This algorithm uses rough set dependency and attributes significance to guide the ants search process and feature gene subset assess. It can get a feature subset easily and quickly, with a small number of genes can get good classification accuracy. The given examples testing in real datasets show that the proposed method is feasible and effective.

Key words: feature selection; rough sets; ant colony algorithm

0 引 言

微阵列实验所检测的基因数量巨大,同时测试费用较高,使得微阵列数据不同于其他的数据载体,具有高维(成千上万)小样本(通常低于一百)的特点,并且存在大量冗余基因与噪声基因。这些特点使得很多传统的分类方法在面对微阵列数据分析时经常表现得很乏力。研究表明,在微阵列数据成千上万的基因中,大部分基因是与分类任务无关的。因此,为了成功地进行分类,选出与分类任务密切相关的基因显得非常重要,这样既可以提高分类精度又有助于降低检测成本,提高疾病预测的可靠度。

蚁群算法是一种启发式优化算法[1],它具有较强的鲁棒性,优良的分布式计算机制,并且易与其他方法相结合。文中结合粗糙集方法和蚁群算法各自的优势,提出一种基于粗糙集和蚁群算法的基因选择方法,将粗糙集的属性依赖度和属性重要度应用到蚁群的路径选择规则中。首先介绍了粗糙集理论的基础,然后由蚁群的特征选择方法开始,循序渐进地引出粗糙集和蚁群算法结合的基因选择方法,对蚁群的路径选择和参数评估等进行了详细的说明。

收稿日期:2011-10-18;**修回日期**:2012-02-02
基金项目:国家自然科学基金(61100116);江苏省自然科学基金(BK2011492);江苏省高校自然科学基金(11KJB520004)
作者简介:黄丹凤(1987-),女,江苏兴化人,硕士研究生,研究方向为计算机应用技术、生物信息学;祁云嵩,教授,博士,硕士生导师,研究方向为模式识别、生物信息学。

1 粗糙集理论

粗糙集(Rough Sets, RS)理论[2,3]是波兰数学家Z. Pawlak在1982年提出的,经过不断的完善与发展,已经形成一套十分完备的理论体系,并被广泛应用于数据挖掘与模式识别等多个领域。它以等价关系为基础,用上下近似两个集合来逼近任意一个集合。粗糙集理论能够解释不精确数据间的关系,发现对象和属性间的依赖,评价属性对分类的重要性,据此对信息系统进行约简。一个信息系统可用一个四元组 $S = < U,$

$A, V, f >$ 表示,其中 U 称为论域,是一个非空有限的对象集合;A 是非空有限的条件属性集合;V 是属性的值域;f 为信息函数,对于 $\forall a \in A, x \in U$,有 $f(x,a) \in V_a$ (V_a 是属性 a 的值域)。$A = C \cup D$,且 $C \cap D = \varnothing$,C 为条件属性集,D 为决策属性集。

定义 1 令 $P \subseteq A$,当 $IND(P) = \{(x,y) \in U \times U \mid \forall a \in P, f(x,a) = f(y,a)\}$,且 $(x,y) \in IND(P)$ 时,称 x 和 y 是 P 不可区分的。

定义 2　对于每个子集 $X \subseteq U$ 和一个等价关系 $R \in IND(P)$,X 的 R 下近似为 $\underline{R}(X) = \cup \{Y \in U/R \mid Y \subseteq X\}$。

定义 3 设 P, Q 为 U 上的两个等价关系簇,Q 的 P 正域定义为 $POS_P(Q) = \underset{X \subset U/Q}{\cup} \underline{P}(X)$。并且,$Q$ 依赖 P 的依赖度定义为 $\gamma_P(Q) = |POS_P(Q)| / |U|$。

定义 4 设 $Q \subseteq P$,如果 $IND(Q) = IND(P)$,称 Q 为 P 的一个约简。

定义 5 条件属性 $a(a \in C)$ 关于决策属性 D 的重要度定义为 $SGF(\alpha, C, D) = \gamma_C(D) - \gamma_{C-a}(D)$。

2　基于粗糙集和蚁群算法的特征基因选择算法

2.1　蚁群算法

研究人员通过观察蚂蚁觅食的行为发现:蚂蚁在前进过程中,会在自己走过的路径上释放出一种化学物质,称之为信息素,其它蚂蚁根据不同路径上信息素的浓度来选择自己所有线路,并在自己选择的路径上释放更多的信息素,某路径上走过的蚂蚁越多,信息素就越浓,该路径便会吸引更多的蚂蚁。蚂蚁间的这种合作使得最终可以得到一条巢穴和食物之间的最短路径[4]。从蚂蚁的觅食过程可以看出,虽然单个蚂蚁是不智能的,但是整个蚁群却能以信息素为媒介进行沟通达到较强的智能。

1991 年,布鲁塞尔自由大学的 Colorni 和 Dorigo 等人受到真实蚂蚁的群体合作行为的启发提出了一种基于群集寻优的启发式搜索算法,即蚁群算法(ACO, Ant Colony Optimization)[5]。目前,该算法已经被广泛应用于多个领域,如数据挖掘[6]、路由分配[7]、蛋白质折叠预测[8]等,并取得了众多成果。

基于蚁群算法的特征选择方法就是通过蚁群的遍历,找到一条从起点到终点的最优路径。文中模型基本思想如下:在基因全集上放入 m 只蚂蚁,每只蚂蚁必须路过所有基因,然后通过蚁群合作决定是否选择该基因。如图 1 所示,每个基因被看作一个结点,每两相邻结点间都有两条路径,分别标记为 0 和 1。蚂蚁必须经过每个结点并选择一个路径,当蚂蚁选择路径 1 时,表示该基因被选择,0 表示未被选择。如一个路径{1,0,1,0,1}表示第 1,3 和第 5 个基因被选取作为特征基因,基因 2 和 4 未被选取。m 只蚂蚁各自遍历一次后便得到了 m 个基因子集。蚂蚁间通过信息素合作,信息素浓度越高,对应路径被选择的概率便越高。当蚂蚁到达食物时,需要用一定的策略对特征子集进行评估选取出最优的特征基因子集。

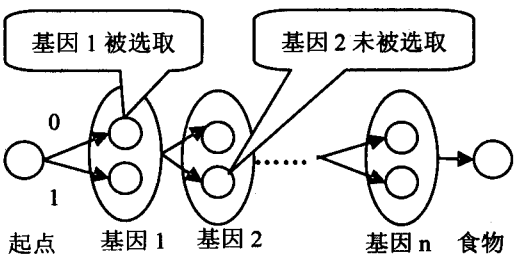

图 1　基于蚁群的特征选择

2.2　算法描述与参数设置

上述模型中有几个问题需要解决:信息素的初始值设定及更新问题,路径被选择的概率设置问题,特征子集质量评估问题。

路径 1 上的信息素初始浓度利用基因 i 关于决策属性 D 的重要度来定义:$SGF(i, C, D)$;路径 0 上的信息素浓度设为 1,这样的作用是路径 0 被选择的概率更高[9],有效控制特征子集长度,使得算法收敛更快,运算更加高效。

路径选择的概率构造如下:

$$P_{ij}^k(t) = \frac{\tau_{ij}}{\sum_j^k \tau_{ij}} \tag{1}$$

其中,τ_{ij} 为第 i 个基因的第 j 条路径上的信息素浓度,k 为 j 的可能取值。

迭代完成后,对所有路径上的信息素浓度更新如公式(2)所示:

$$\tau_i(t+1) = (1-\eta) \times \tau_i(t) + \Delta\tau_i(t) \tag{2}$$

其中,η 为信息素的蒸发因子,模拟了蚂蚁分泌的信息素在自然环境下不断挥发导致信息素浓度随着时间的推移而逐渐衰减的效果[10];$\Delta\tau_i(t)$ 为信息素的增量,如公式(3)所示:

$$\Delta\tau_i(t) = \begin{cases} 1/(\theta \times L(k) + (1-\theta) \times (\gamma(k))^{-1}), & \text{pathway} \in S \\ 0, & \text{pathway} \notin S \end{cases} \tag{3}$$

上式中,$L(k)$ 表示蚂蚁 k 搜索到的特征子集长度,即路径中 1 的数目。$\gamma(k)$ 表示决策属性对该特征子集的依赖度。$\theta \in (0,1)$,用来控制特征子集的长度和性能。为了减少计算量,并更好地区分不同路径上的信息素浓度,S 取前 10% 的蚂蚁对应的路径[9]。$L(k)$ 越小,$\gamma(k)$ 越大,则信息素增量越大,将来蚂蚁

选择该路径的可能性也越大，这样可使得特征子集朝着最短，决策属性依赖度最大的方向发展。

当蚂蚁到达食物时，将结合特征子集长度和属性依赖度来评估对应的特征基因子集的质量，定义如下：

$$\varepsilon(S) = n/(\theta \times L(S) + (1-\theta) \times (\gamma(S))^{-1}) \tag{4}$$

其中，n 为数据集的基因数。基于该评价函数可以找出决策属性依赖度最大的最短特征基因子集。

2.3 算法描述

算法：基于粗糙集和蚁群算法的特征基因选择方法

输入：数据集 $T = (U, A, V, f)$，算法各参数。

输出：特征子集 CS。

步骤 1：信息初始化：设置初始时刻各特征节点上的信息素浓度 $\tau_i(0)$，最大迭代次数 W_{max}；

步骤 2：每只蚂蚁根据公式(1)选择一条从巢穴到达食物的路径，构造特征基因子集；

步骤 3：利用公式(4)对每个特征子集进行评估，并将前 10% * m 只蚂蚁对应的路径存到 S 中供信息素更新参考。选取本次最优解，将本次最优解与以前得到的最优解进行比较，选择较优的；

步骤 4：如果最优基因子集没有满足要求并且迭代次数小于最大迭代次数 W_{max}，按公式(2)和公式(3)的方法更新信息素浓度，返回步骤 2 继续运行；否则输出特征子集 CS。

3 仿真实验与结果分析

3.1 数据集

如表 1 所示，文中采用两种典型的数据集：白血病数据集和结肠数据集。

表 1 实验数据集

数据集	特征基因总数	样本个数	类别个数
白血病数据集	7129	72	2
结肠数据集	2000	62	2

白血病数据集是 Golub 等人在 1999 年给出的[11]，它共有 72 个样本，其中包括 47 个为急性淋巴白血病(ALL)样本，25 个急性骨髓白血病(AML)样本，每个样本含有 7129 条基因。数据获取途径：http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi。

结肠数据集是 Alon 等人收集的数据[12]，它共有 62 个样本，其中包括 40 个肿瘤样本，22 个正常结肠组织样本，每个样本含 2000 条基因。数据获取途径为：http://www.molbio.princeton.edu/colondata。

3.2 实验环境与结果分析

软件：操作系统 Windows XP，编译软件 Matlab7.1；使用 Matlab 自带的支持向量机(SVM)分类器。

硬件：Pentium(R) Dual-Core CPU 2.61 GHz，内存 2GB。

具体参数取值如下：

蚂蚁数量 $m = 50$，最大迭代次数 $W_{max} = 50$；蒸发因子 $\eta = 0.5$，参数 $\theta = 0.2$。

首先利用 RS 理论从两个数据集上初选出前 100 个与样本分类密切相关的基因测试分类准确率。样本具体分配如下：白血病数据集训练集 36 个，其中 23 个 ALL 样本为正样本，13 个 AML 为负样本；测试集 36 个，ALL 和 AML 样本数分别为 24，12。结肠数据集训练集 31 个，正样本和负样本数分别为 20，11；测试集分布相同。支持向量机分类结果为：白血病数据集上的分类准确率为 86.1%，结肠数据集上为 83.9%。然后采用文中的算法先在这些基因中选出最优特征基因子集再查看分类效果，特征子集规模及分类准确率如表 2 所示。可以看出文中算法可以获得较小规模的特征基因子集，分类性能也得到有效提高。以白血病数据集为例，只需要迭代 16 次便可获得只有 7 个基因的最优解，分类准确率也得到了提高。

表 2 特征子集规模及分类结果

数据集	特征子集规模	分类准确率
白血病数据集	10	94.4%
结肠数据集	7	90.3%

4 结束语

文中结合粗糙集与蚁群算法的优势提出了一种基因选择方法，将粗糙集理论中的属性依赖度和重要度引入蚁群算法来进行路径的选择和特征子集的评估，最终获得最优特征基因组合。优化了算法的性能，实现简单，可以较快速地获得一个最优特征基因组合。实验表明，该方法是可行的，通过文中提出的算法，能够较好地选择出最具有表达性的特征基因，获得较好的分类效果。

从基因选择的实验过程中也发现，算法易于陷入局部最优解，如何合理高效地处理这一问题，将是下一步的研究重点。

参考文献：

[1] Dorigo M, Maniezzo V, Coloni A. Ant system: optimization by a colony of cooperating agents[J]. Institute of Electrical and Electronics Engineers Transactions on Systems, Man and Cybernetics, 1996, 26(1): 8-41.

[2] 苗守谦，李道国. 粗糙集理论、算法与应用[M]. 北京：清华大学出版社，2008.

(下转第 74 页)

2.5 织入器

织入的实现机制有多种，从织入的过程看可以分为两类：静态织入与动态织入。本平台采用静态织入方法，即在核心功能代码中的适当位置，比如某段代码执行前，或执行后，将 Aspect 代码织入，从而形成混合的编码。即调用 ajc 编译器，它把 Aspect 特有的关键字编译成字节码织入到主应用程序字节码中，并产生可由 Java 运行环境来执行的、适当的 aj 文件。

3 结束语

构件测试是近年来受到软件测试界普遍关注的一个话题，而构件内部信息屏蔽、演变速度较快以及构件间的异质、松耦合等特点给软件构件系统的测试带来了极大不便。文中提出的基于 AOP 的构件合约测试方法，将合约代码当作横切关注点，与具体的业务逻辑代码（核心关注点）进行了有效分离。同时，基于此方法实现的构件测试平台通过合约编辑器添加合约内容，使代码编写者不必关注合约代码的语言规范和书写形式，降低了代码编写难度，减轻了开发人员的工作量。

如果程序在运行中违背了合约，就需要根据具体情况产生不同的错误提示信息，进行相应处理。文中实现的测试平台仅仅是个原型，只是简单记录了错误信息的相关内容。下一步的工作重点是将原型平台变为实用程序平台。

参考文献：

[1] Vincenzi A M R, Maldonado J C, Wong W E, et al. Coverage testing of Java programs andcomponents[J]. Science of Computer Programming, 2005, 56(1/2): 211-230.

[2] Mayer B. Applying" Design by Conrtact" [J]. IEEE Computer, 1992, 25(10): 40-51.

[3] Gao J Z, Tsao H S J, Wu Y. Testing and Quality Assurance for Component-based Software[M]. Boston: Artech House, 2003.

[4] Cheng Y C, Chen Chien-Tsun, Hsieh Chin-Yun. ezContract: Using Marker Library and Bytecode Instrumentation to Support Design by Contract in Java[C]//Software Engineering Conference. [s. l.]: [s. n.], 2007: 502-509.

[5] 樊庆林，吴建国. 提高软件测试效率的方法研究[J]. 计算机技术与发展，2006，16(10)：52-54.

[6] Feldman Y A, Barzilay O, Tyszberowicz S. Jose: Aspects for Design by Contract[C]//Proceedings of the Fourth IEEE International Conference on Software Engineering and Formal Methods(SEFM'06). [s. l.]: [s. n.], 2006.

[7] 赵艳妮，王映辉，雷　宇. 一种基于 AOP/IOC 的软件框架研究与实现[J]. 计算机工程与应用，2008，44(29)：92-95.

[8] 陈　成，李　行. 基于 AOP 的 MDA 模型转换[J]. 计算机技术与发展，2008，18(7)：87-89.

[9] 古全友，王恩波，胥昌胜. AOP 技术在 J2EE 系统构建中的应用[J]. 计算机技术与发展，2006，16(4)：150-152.

[10] 李志纯，张南平. 面向 Aspect 编程的应用研究[J]. 计算机技术与发展，2006，16(5)：217-220.

[11] Graddeck J D, Lesiecki N. 精通 AspectJ[M]. 北京：清华大学出版社，2004.

[12] 周庆泉，郝克刚. 基于 AOP 的工厂模式研究[J]. 计算机技术与发展，2008，18(8)：47-49.

[13] 尹　涛，李　翔，林　祥. 基于 AOP 的角色访问控制模型设计与实现[J]. 计算机技术与发展，2008，18(10)：136-142.

（上接第 70 页）

[3] Pawlak Z. Rough sets-theoretical aspects of reasoning about data[M]. [s. l.]: Kluwer Academic Publishers, 1991.

[4] 马　良，朱　刚，宁爱兵. 蚁群优化算法[M]. 北京：科学出版社，2008.

[5] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]//Proceedings of 1st European Conference on Artificial Life. Paris, France: Elsevier Publishing, 1991: 134-142.

[6] Parpinelli R S. Data mining with an ant colony optimization algorithm[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 321-332.

[7] Caro G D, Dorigo M. AntNet: distributed stigmergetic control for communications networks[J]. Journal of Artificial Intelligence Research, 1998(9): 317-365.

[8] Shmygelska A, Hoos H H. An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem[J]. BMC Bioinformatics, 2005(6): 30-30.

[9] 于化龙. 基于 DNA 微阵列数据的癌症分类技术研究[D]. 哈尔滨：哈尔滨工程大学，2010.

[10] 蔡立军，蒋林波，易叶青. 基于蚁群优化算法的基因选择[J]. 计算机应用研究，2008，25(9)：2754-2757.

[11] Golub T R, Slonim D K, Tamayo C P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring[J]. Science, 1999, 286(15): 531-537.

[12] Alon U, Barkai N, Notterman D A, et al. Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues by Oligonucleotide Arrays[J]. Proceedings of the National Academy of Science, 1999, 96(12): 6745-6750.