基于基因表达式编程的推进学习算法

罗 谦[1],左 桃[2]
(1. 中国民用航空总局第二研究所 信息公司,四川 成都 610047;
2. 中国人民银行 成都分行,四川 成都 610041)

摘 要:为进一步改善传统遗传算法在函数发现应用上的搜索能力,提高算法的收敛速度和精度,文中提出了 GEPAdaBoost 算法。该算法在推进学习 AdaBoost 算法框架下,利用具有强大函数发现能力的基因表达式编程 GEP 作为每次迭代过程中的弱学习器,同时引入含分布因子的适应度函数在迭代中筛选出最优假设,最后通过投票策略组合多轮最优假设产生算法结果。丰富的实验结果表明新算法对权重计算和概率分布产生了积极的影响,与朴素 GEP 算法和 GPBoosting 算法对比分析发现该算法能分别提升 16.7% 和 40.8% 的精度。

关键词:遗传算法;基因表达式编程;机器学习;推进学习
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2013)02-0165-05
doi:10.3969/j.issn.1673-629X.2013.02.040

AdaBoost Algorithm Based on GEP

LUO Qian[1],ZUO Tao[2]
(1. Information Filiale,The Second Research Institute of CAAC,Chengdu 610047,China;
2. Chengdu Branch Bank,People's Bank of China,Chengdu 610041,China)

Abstract:In order to improve the tradition GA's searching ability in regression problems,enhance the algorithm in convergence rate and precision,propose GEPAdaBoost algorithm. Based on the frame of AdaBoost,the GEPAdaBoost takes GEP as a weak-learner for every iteration by using its power ability of symbolic regression. Then the new fitness function of every iteration can produce good hypothesis based on weight computing and distributing. Finally the algorithm will get optimization result by using voting strategy in multi- hypothesis. Experiments show that the new algorithm is more accurate than the traditional GEP algorithms by 16.7% and GPBoosting algorithms by 40.8%.

Key words:genetic algorithms;gene expression programming (GEP);machine learning;boosting

0 引 言

从遗传学角度来看,种群的进化需要在尽可能保证物种多样性(随机变异)的前提下,利用“优胜劣汰”法则(适应度函数),繁衍出最适合生存的物种(最优结果)。整个进化过程中,物种将被多次划分可以遗传和直接抛弃的两大类,这是一个典型的二分类问题。该问题的求解质量与否直接关系了最优结果产生的质量和速度。

遗传算法(GA)和遗传编程(GP)从不同的角度利用上述的基本原理解决了很多优化问题:如复杂多目标规划问题,配管、配线问题,神经网络权系数调整和网络构造等问题,但在求解过程中又暴露了诸多不足:如局部搜索能力差、早熟和随机漫游等等。基因表达式编程(Gene Expression Programming,GEP)[1]则是遗传算法(GA)和遗传编程(GP)融合的产物,它较好解决了算法收敛速度问题。GEP 在解决复杂问题上比传统的遗传编程方法效率高出 2~4 个数量级[2],但在上述的二分类问题的求解过程 GEP 并没有带来本质的变化。

机器学习领域的 Boosting 算法则在每次迭代中,基于训练样本集产生多个假设,通过计算错误率在训练样本集上维护一套概率分布,以多个单分类器的加权投票建立最终分类器,从而获得足够的预测精度。1989 年 Schapire 证明了在 Valiant 的 PAC 模型中,一个“弱”学习算法能被“提升”为一个具有任意精度的“强”学习算法[3]。

文中旨在融合基于遗传进化理论的 GEP 与机器学习的 Boosting 算法,在 Boosting 算法的每次迭代中引

收稿日期:2012-06-06;**修回日期**:2012-09-12
基金项目:国家科技支撑计划课题(2012BAG04B02);中国民用航空局科研项目(MHRD200924)
作者简介:罗 谦(1975-),男,四川宜宾人,博士,主要研究方向为数据挖掘、进化计算、企业智能计算。

入 GEP 作为其函数发现的分类器,多次迭代后能获得一个较单纯使用二者更优的结果。这样既保证了产生分类器的多样性(没有丢弃,强制保留),又确保了重点突出分类器的作用(权重的不同)。通过实验证明,GEPAdaBoost 算法较朴素的 GEP 算法和 GPBoosting 算法性能提高了 16.7% 和 40.8%。

1 相关研究内容

研究界将进化算法(Evolutionary Algorithms)和合成学习(Ensemble Learning)的结合问题归结为多目标优化问题(Multiple Objection Optimization),即获得最小化错分率和最大化多样性结果。文献[4]在进化过程中,利用基于后向传播算法的神经网络(Back-Propagation)保证了物种的多样性,同时最小化了训练集的错误;同样的目标在 Chandra 和 Yao 的 DIVACE 系统[5,6]得以实现,文献[6]中高层进化(Top-level evolution)思想确保最小化了错误率和最大化了负相关系数(Negative Correlation);而文献[5]则利用 Pairwise Failure Crediting 替代了负相关系数(Negative Correlation),更有效表达了错分的样本被其他分类器准确分类的情况。其他的研究热点就把多目标优化问题分而治之,有的直接把进化算法应用到标准的 Bagging 和 Boosting 算法中,并实现了分类应用[2]和函数发现[7];有的则是将一个很大的种群划分为若干个小的种群,再进行计算[8];Folino 等是利用并行细胞 GP 来搭建自己的算法框架[9];Song 等使用类似于 Boosting 的启发算法[10]解决了种群数目过大需要实现内存数据交换的问题。但到目前为止,还没有看到有关 Boosting 算法和 GEP 相结合的研究成果。

2 基本概念和术语

推进学习 BoostingLeaning 算法思想由 Freund 和 Schapire 于 1990 年提出,是提高预测学习系统预测能力的有效工具,也是组合学习中最具有代表性的方法。其中 PAC 模型[11]涉及 2 个概念:强学习和弱学习。

定义 1 (强弱学习)设 S 为含 N 个数据点 (x_1,y_1), $\cdots$, (x_N,y_N) 的样本集,其中 x_N 是按照某种固定但未知的分布 $D(x)$ 随机独立抽取的。设布尔函数集 $Y_k = f(x_k), k = 1,2,\cdots,n, 0 < k < n+1$。任意给定 $D, f \in F$,和 ε,σ,满足 $0 \leqslant \varepsilon,\sigma \leqslant 1/2$。则满足下列两个条件中(a),(b)或(a),(c)的成为弱学习算法,满足全部三个条件的称为强学习算法。

(a) A 从 S 中学习除模式 P, d 使得 $Pr[h(x) \neq f(x)] \leqslant \varepsilon$ 的估计 h 的概率大于 $1-\sigma$;

(b) A 相对于 $1/\varepsilon$, 具有多项式计算复杂度 $p(1/\varepsilon)$;

(c) A 相对于 $1/\sigma$,具有多项式计算复杂度 $p(1/\sigma)$。

1995 年,Freund 和 Schapire 提出了 Adaboost(Adaptive Boosting)算法[3],这种算法的效率和原来 Boosting 算法的效率一样,但不需要任何关于弱学习器性能的先验知识,因此可以非常容易地应用到实际问题中。

可以证明,AdaBoost 调用给定的弱学习算法 WeakLearn 时,将产生错误率为 $\varepsilon_1,\cdots,\varepsilon_T$ 的假设。假设每个 $\varepsilon_t \leqslant 1/2$,于是最终假设 $F(x)$ 的错误 $\varepsilon = P_{\tau_t \sim D}[h_t(x_t) \neq y_t]$ 的上边界为[12]:

$$\varepsilon \leqslant 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t(1-\varepsilon_t)}\ 。$$

而 2001 年 12 月,Candida Ferreira 在遗传算法[1]的基础上提出了基因表达式编程(Gene Expression Programming,GEP)的概念[2],GEP 与传统的遗传算法以及遗传编程(GP)在一些主要步骤上很相似,但在个体的基因型的编码方法及个体的表现型等方面又存在明显的区别。

它们最本质的区别在于:在 GA 中个体由固定长度的线性串(染色体)来表示;在 GP 中个体由不同大小和形状的非线性实体(解析树)所表示;而 GEP 将个体先编码为固定长度的线性串,再表示成大小、形状都不同的非线性实体。GEP 是 GA 和 GP 的继承和发展,它综合了 GA 和 GP 的优点,具有更强的解决问题的能力[2]。

例 1:考虑 Sin((a/b - c)(a + d))。$F = \{S, +, -, *, /\}$ (S 表示 Sin), $V = \{a,b,c,d\}$ Gene 的 |head| = 5。对该表达式产生的 ET(表达式树)进行层次遍历可以得到序列:"S * -+/cadab",这个式子又称为 K-表达式(K-expression)[2],它构成了数学表达式的基因编码的主体。此例中,编码只有前 10 位被算子解析使用,余下的位可以填充任意变量。

GEP 作为一种通用的自适应式随机搜索算法,在很多应用领域取得了很好的实际效果,尤其在符号回归问题(函数发现)具有强大的表达式编程能力和优异性能,关于 GEP 的研究和应用参见文献[13,14]。

3 基于推进学习的 GEP 算法(GEPAdaBoost)

为了解决物种多样性和优胜劣汰矛盾问题,融合 Boosting 算法和 GEP 的二者的优点,提出了 GEPAdaBoost 算法。有下列特点:

(1)大框架沿用 AdaBoost 算法;

(2) 在每次迭代过程中利用了 GEP 符号回归的高速度和对复杂函数予以的表达能力,作为弱学习器

求其假设 h_i;

(3)每次迭代利用新的适应度函数筛选出当前最优的假设进入最后的投票决策中;

(4)组合获得最终假设。

实验表明该算法能以更高的精度解决传统的函数发现问题。

算法 1(GEPAdaBoost):基于 AdaBoost 推进学习的 GEP 算法。

输入:N 个带标记实例的序列集合 $R=\{(x_1,y_1),\cdots,(x_n,y_n)\}$,这 N 个实例上的分布为 D,学习工具为朴素 GEP,迭代次数为 T。

输出:$F(x)$。

步骤:

//初始化权重向量

1　for　$i=1\cdots N$

　　$D[i]=1/N$; //一般情况下初始分布一致

　end for

2　for t=1…T

　　//调用学习器朴素 GEP

3　h_t =GetHyphBasedonGEP($D[1\cdots N]$);

　　//调整分布

4　AdjustDistribution ($D[1\cdots N]$, h_t);

5　end for;

　　//其中 x 来自 R

6　$F(x)=$

$$\begin{cases}1 & \text{若}\sum_{t=1}^{T}(\log\frac{1}{\beta_t})h_t(x)\geqslant\frac{1}{2}\sum_{t=1}^{T}\log\frac{1}{\beta_t}\\ 0 & \text{其他}\end{cases}$$

7　return $F(x)$

算法中的两个核心函数如下:

●Function GetHyphBasedonGEP ($D[1\cdots N]$)

1　…//经典的朴素 GEP 算法

2　gep. run( );

3　Fitness= $\sum_{i=1}^{N}(P_i^t\mid f(x_i)-y_i\mid *D_t(i))*N$

　//返回 GEP 进化出来的最佳假设函数

4　return Max(h_t);

●Function AdjustDistribution ($D[1\cdots N]$)

　//计算 h_t 的错误

1　$\varepsilon_t=\sum_{i=1}^{N}P_i^t\mid h_t(x_i)-y_i\mid$

2　$\beta_t=\varepsilon_t/(1-\varepsilon_t)$;

　//计算新的权重向量

3　$D[i]^{t+1}=D[i]^t\beta_i^{1-|h_t(x_i)-y_i|}$

4　return $D[1\cdots N]$

较传统的 AdaBoost 算法,新的 GEPAdaBoost 算法主要涉及以下两个方面的讨论。

(1)GEP 与弱学习器。

结合定义 1,可以看出理论上 GEP 并不符合这两个定义。因此,把 GEP 作为弱学习器是一种尝试,是一种对试验结果的观察,无法从理论上保证 GEP 同其他弱学习器一样确保错误率的有效降低。但文献[2,7,15]中利用 GP 作为弱学习器的所有试验表明 GP 能有效降低错误率。因此,完全可以尝试把 GEP 作为一种弱学习器。

(2)含分布因子的适应度函数。

将 GEP 蕴含在 AdaBoost 算法框架中,作为每一次迭代的分类器,那么传统 GEP 适应度函数则无法满足 AdaBoost 算法涉及分布的需求,会丢失每次迭代所带来的分布变化值。因此,在 GEP 基于绝对误差适应度函数的基础上引入了对概率分布的影响,才能更有效表达 GEPAdaBoost 算法的精度。

传统的 GEP 适应度函数主要是基于绝对误差和相对误差两种形式。

$$f_i=\sum_{j=1}^{C_t}(M-|C_{(i,j)}-T_j|) \tag{1}$$

$$f_i=\sum_{j=1}^{C_t}\left(M-\left|\frac{C_{(i,j)}-T_j}{T_j}.100\right|\right) \tag{2}$$

融入分布因子后的 GEPAdaBoost 算法适应度函数:

$$f_i=\sum_{i=1}^{N}(P_i^t-|h_t(x_i)-y_j|)*N \tag{3}$$

4　实验及性能评估

4.1　实验环境

文中提出的 GEPAdaBoost 算法在以下实验环境完成:CPU:Xeon 2×1.8GHz;内存:8G;硬盘:2×143G;操作系统:Red Linux 企业版;Java Jdk1.6;IDE:Eclipse3.5;Weka3.6.5。

为了进一步对比说明 GEPAdaBoost 算法的高效性,文中重写了文献[15]的 GPBoosting 算法,相关参数设置与文献[15]一致。两个算法均基于 Weka.meta 中 AdaBoost.M1 框架。

4.2　实验 1

实验 1 模拟了一元函数。

$$y=x^2/2 \tag{4}$$

在实验中,首先指定训练数据集(参见表 1);然后每一次迭代都用 GEP 在训练集上产生最优的假设函数 $f_i(x)$;同时修正新的概率分布 D_i;迭代完成后投票产生最终的假设函数组合。每次迭代过程中的 GEP 适应度函数如式(1)所示。GEP 具体参数设置如表 1 所示,表 2 为外层 Adaboost GEP 的参数设置。

表 1 实验 1 中的 GEP 参数

选项	参数
运行次数	1
最大进化代数	150
种群数量	30
符号集	+,-,*,/
终结符	X
连接函数	+
头长	7
基因组个数	3
变异率	0.044
染色体重组率	0.044
基因迁移率	0.3
适应度函数	式 1 所示

表 2 实验 1 中的 AdaboostGEP 参数

选项	参数
终结符	X
目标函数	$y=x^2/2$
样本空间	{(-0.1,0.005),(-0.5,0.125),(1,0.5),(0.2,0.02),(0.7,0.245)}
弱学习器	GEP
迭代次数	5

图 1 实验结果(适应度值同比缩小 10 倍)表明朴素 GEP 和 GEPAdaBoost 均能在 100 代 ~ 150 代进化过程中,挖掘出 $y = x^2/2$。但朴素 GEP 的挖掘效率较 GEPAdaBoost 略高,这是由于在简单函数发现过程中,GEPAdaBoost 的学习框架平均了误差造成的。而 GPBoosting 算法求解精度分别较朴素 GEP 和 GEPAdaBoost 低 33.5% 和 27.4%。

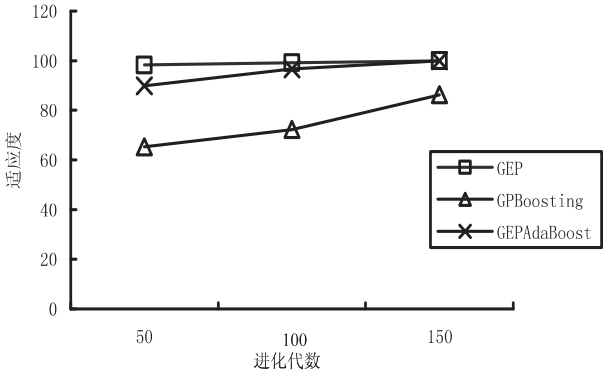

图 1 实验 1 的平均适应度

4.3 实验 2

实验 2 则模拟了复杂的一元多次函数。

$$y = x^3 e^{-x} \cos x \sin x \,(\sin^2 x - 1) \tag{5}$$

实验中首先产生 200 个[-1,+1]间的随机数作为 x 值送入式(5)产生对应的 y 值,这 200 个数据对构成训练集。然后每一次迭代都用 GEP 在训练集上产生最优的假设函数 $f_i(x)$;同时修正新的概率分布 D_i;迭代完成后投票产生最终的假设函数组合。每次迭代过程中的 GEP 适应度函数如式(3)所示。GEP 具体参数设置如表 3 所示,表 4 为外层 Adaboost GEP 的参数设置。

表 3 实验 2 中的 GEP 参数

选项	参数
运行次数	1
最大进化代数	500
种群数量	100
符号集	+,-,*,/,S,C,E,L
终结符	X
连接函数	+
头长	8
基因组个数	3
变异率	0.044
染色体重组率	0.044
基因迁移率	0.3
适应度函数	式 3 所示

其中函数符集中的 S、C、E、L 分别为 Sin、Cos、Exp 和 Log 运算符。

表 4 实验 2 中的 AdaboostGEP 参数

选项	参数
Terminal	X
目标函数	$y=x^3 * e^{-x} * \cos(x) * \sin(*) * [\sin^2(x) * \cos(x) - 1]$
样本空间	{200 个随机产生的(x,y),其中 $x \in [-1,+1]$}
弱学习器	GEP
迭代次数	10

图 2 实验结果(适应度值同比缩小 100 倍)表明在复杂函数的发现过程中,GEPAdaBoost 精度最高,较朴素 GEP 求解精度提高了 16.7%,较 GPBoosting 算法求解精度提高了 40.8%。与简单函数发现相比,GEPAdaBoost 算法表现出随着函数的复杂度提升其精度越高的趋势。

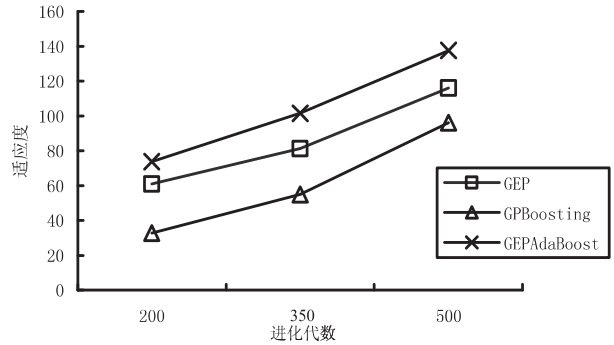

图 2 实验 2 的平均适应度

5 结束语

从上述可以看出,GEPAdaBoost 算法在 AdaBoost 迭代实现框架下强制保留了物种样本的多样性,并且在每一次迭代中充分利用了 GEP 符号回归的高速度和对复杂函数予以的表达能力,形成有效假设。在基于绝对误差和相关系数共同影响下的权重数值,形成了下一次迭代的概率分布,最终假设仍然采用投票策略。

实验表明,GEPAdaBoost 算法效率较传统的 GEP

算法提高了 16.7%;较 GPBoosting 提高了 40.8%。在下一步的研究工作中将重点考虑 GEPAdaBoost 算法对噪声敏感的问题,并打算将 GEPAdaBoost 应用在银行信用欺诈检测中,当然对最终假设的其他组合选择也是关心的内容。

参考文献:

[1] Ferreira C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems[J]. Complex Systems,2001,13(2):87-129.

[2] Ferreira C. Gene Expression Programming:Mathematical Modeling by Artificial Intelligence[M]. Portugal:[s. n.],2002:146-151.

[3] 涂承胜,刁力力,鲁明羽,等. Boosting 家族 AdaBoost 系列代表算法[J]. 计算机科学, 2003,30(3):30-34.

[4] Liu Y,Yao X,Higuchi T. Evolutionary ensembles with negative correlation learning[J]. IEEE Trans. on Evolutionary Computation,2000,4(4):380-387.

[5] Chandra A,Yao X. Ensemble learning using multi-objective evolutionary algorithms[J]. J. of Mathematical Modeling and Algorithms,2006,5(4):417-425.

[6] Chandra A,Yao X. Evolving hybrid ensembles of learning machines for better generalization[J]. Neurocomputing,2006,69(7-9):686-700.

[7] Keijzer M,Babovic V. Genetic programming,ensemble methods,and the bias/variance tradeoff - introductory investigations[C]//Proc. of the EuroGP'00. Berlin:Springer,2000.

[8] de Souza L V,Pozo A T R,Neto A C. Using Correlation to Improve Boosting Technique: An Application for Time Series Forecasting[C]//ICTAI'06. Arlington:IEEE,2006.

[9] Folino G,Pizzuti C,Spezzano G. Ensemble techniques for parallel genetic programming based classifiers[C]//Proc. of EuroGP'03. Berlin:Springer,2003.

[10] Song D,Heywood M I,Zincir-Heywood A N. Training genetic programming on half a million patterns:an example from anomaly detection[J]. IEEE Trans. on Evolutionary Computation,2005,9(3):225-239.

[11] 于　玲,吴铁军. 集成学习:Boosting 算法综述[J]. 模式识别与人工智能,2004(1):52-59.

[12] 涂承胜,陆玉昌. Boosting 理论基础[J]. 计算机科学,2004,31(10):11-14.

[13] Ferreira C. Discovery of the Boolean Functions to the Best Density-classification Rules Using Gene Expression Programming[C]//Proc of EuroGP 2002. Berlin:Springer,2002.

[14] Ferreira C. Mutation,transposition and recombination:An analysis of the evolutionary dynamics[C]//4th Int'l Workshop on Frontiers in Evolutionary Algorithms. North Carolina, USA: Research Triangle Park,2002:614-617.

[15] Paris G,Robilliard D,Fonlupt C. Applying Boosting techniques to genetic programming[C]//AE2001. Paris:Springer,2002.

(上接第 164 页)

化的开发工具将为开发者提供极大帮助,所有基础的细节自动工具都将为开发者生成,同时将联动开发者任何的修改,而开发者只需将注意力集中到业务逻辑本身;完整的基础支持将为软件项目开发提供定制化的能力,平台为应用提供了工作流、报表等各种技术资源,应用程序只要根据需要进行组合和简单的二次开发就可以应用到软件项目上,为软件项目减少了大量的开发成本。目前,泰州某大桥建设工程项目系统,某市地铁工程建设项目系统等多个大型工程建设项目系统,均在此平台上开发。

实际结果证明,有了通用平台,加快了项目实施速度,获得了很好的用户评价,为企业争取了更多的效益。后期可以在提供系统框架的规模化能力、通用业务组件的丰富,建设项目的标准化规范、为中大型项目提供全面的企业级解决方案等方面做些工作。

参考文献:

[1] 常　盛,郑世钰. 通用型信息管理系统的探索与研究[J]. 硅谷,2011(24):82-82.

[2] 王　敏,陈亚光. 通用信息管理系统设计模式[J]. 现代科学仪器,2009(2):3-6.

[3] 谢　辉,魏金岭,马　楠. 通用标准化高校教学管理信息系统分析与设计[J]. 计算机系统应用,2009(10):24-26.

[4] 吴　江,王铭叶. 基于通用设计的信息产品界面设计研究[J]. 包装工程,2009(12):225-227.

[5] 任　斐. 基于 J2EE 通用信息平台的关键技术[J]. 信息与电子工程,2006(2):134-137.

[6] Su Anyu,Liu Xiaoguang. Research of GIS Platform Based on MapX[J]. Journal of Northeast Agricultural University (English Edition),2008(2):88-91.

[7] Memorandum of Sino-EU/EFTA Cooperation in Standardization Information Platform Signed[S]. 2010.

[8] 俞　晓,苗　放. Information System for Land-use Planning and Management[J]. Journal of Southwest Jiaotong University (English Edition),2008(4):22-24.

[9] Guo Hua. A Sustainable Platform for E-Service System Design [J]. Journal of Systems Science and Systems Engineering, 2004(4):78-81.

[10] 傅文博. 通用管理信息系统开发平台的构建方法及实现[J]. 软件导刊,2009(11):123-124.

[11] 徐毅靖. 国土资源电子政务通用平台开发及应用[J]. 地理信息世界,2005(6):36-40.

[12] 张绍缔. 通用信息管理系统开发平台的设计与实践[J]. 信息技术与信息化,2011(5):32-33.

基于基因表达式编程的推进学习算法

作者： 罗谦， 左桃
作者单位： 罗谦(中国民用航空总局第二研究所 信息公司,四川 成都610047)， 左桃(中国人民银行成都分行,四川 成都610041)

刊名： 计算机技术与发展
英文刊名： Computer Technology and Development
年，卷(期)： 2013(2)