用实数编码的遗传算法构造斜决策树*)

Oblique Decision tree Construction with Decimal-coded Genetic Algorithm

胡宏银　朱绍文　张大斌　王泉德　黄　浩　陈菁华　陆玉昌
(华中师范大学电子与计算机研究所　武汉 430070)(清华大学计算机系　北京100084)

Abstract The algorithm finding the coeffcients of linear combinated attributes with decimal-coded genetic algorithm and constructing decision tree is explored in this paper, and the results of experiment show this algorithm is efficient and effective.

Keywords Data mining, Oblique decision tree, Linear combinated attributes, Genetic algorithm

决策树方法是一种通过构造决策树来发现训练集中分类知识的数据采掘方法，其核心是如何构造决策树，构造决策树的关键是找出表示内部节点的最佳扩展属性[1]。扩展属性有单属性和联合属性，由单属性形成的扩展属性集小，可以容易地找出最佳扩展属性，构造单元树的速度快，但是生成的单元树规模大，并可导致子树复制、一个属性的多次测试等；用联合属性作为扩展属性，生成的多元树规模小，能有效地克服单元树存在的问题，因此，多元树的构造受到许多研究者的重视。多元树中研究得最多的是斜决策树，它的内部节点为线性联合属性，寻找最佳线性联合属性的过程是不断优化属性系数的过程，优化系数的方法有爬山搜索法[2,3]、模拟退火法[4]等。

遗传算法是一种新兴的并行搜索寻优技术，它仿效生物的进化与遗传，根据"生存竞争"和"优胜劣汰"的原则，借助复制、杂交、突变等操作，使所要解决的问题从初始解一步步地逼近最优解。研究表明，遗传算法是一种良好的优化方法，并且已经应用于很多领域。本文设计一种实数编码的遗传算法，用它来优化线性联合属性的系数，找出最佳的线性联合属性，并进一步构造斜决策树。

1　线性联合属性

设 $E=\{e_i|i=1,\cdots,n\}$ 是决策树的训练集，e_i 称为事例，$e_i=\{a_j=x_{ij}|j=1,\cdots,r\}\cup\{a_L=x_{iL}\}$；$A=\{a_i|i=1,\cdots,r\}\cup\{a_L\}$ 是属性集，其中 $a_j,j=1,\cdots,r$ 称为描述属性，a_L 称为类别属性；$X_j=\{x_{ij}|i=1,\cdots,n\}$ 是描述属性 a_j 的属性值集，$X_L=\{x_{iL}|i=1,\cdots,n\}$ 是类别属性 a_L 的属性值集，即类别集。事例 $e_i(e_i\in E)$ 的类别是 $x_{iL}(x_{iL}\in X_L)$，记为 $x_{iL}=f(e_i)$。事例可看作是 r 维描述属性空间中的点，点的特性为该事例的类别属性值，训练集是空间中的点集。线性联合属性的形式为：

$$\sum_{i=1}^{r} w_i a_i + w_{r+1}$$

其中系数 $w_i\in R$，且 $-1\leqslant w_i\leqslant 1, i=1,\cdots,r,r+1$。线性联合属性表示用超平面 $\sum_{i=1}^{r} w_i a_i + w_{r+1}=0$ 把点集划分为两部分，系数决定着超平面的位置和方向。线性联合属性是对数值型属性而言的，离散型属性需先数值化，本文假定 r 个描述属性都为数值型属性。

2　用遗传算法优化线性联合属性的系数

用遗传算法优化线性联合属性的系数，编码是基础，常见的编码方法有二进制编码和实数编码。二进制编码技术通用性强，操作简单，适用于离散型优化变量或数目较少的连续型优化变量；但是，线性联合属性的系数为连续型优化变量，且数量较多，二进制编码会使码串过长，算法收敛速度慢，并且操作时需要不断地编码和译码，既增加了计算量，又损失了精度。本文采用实数编码的遗传算法，直接用系数表示基因，系数按顺序组成染色体(个体)$w=w_1w_2,\cdots,w_rw_{r+1}$，记为 $w=(w_1,w_2,\cdots,w_r,w_{r+1})$。个体通过不断地进行复制、杂交、变异等遗传操作，找出全局最优系数。

2.1　复制

复制是根据个体适应度的大小选择优良个体在下一代群体中繁殖。

定义1　设数据集 $E'\subseteq E, L'=\{f(e)|e\in E'\}$，若 $|L'|=1$，则称 E' 是纯数据集，否则称 E' 是不纯数据

*)清华大学技术智能与系统国家重点实验室开放课题资助项目。朱绍文　教授，主要研究方向为数据采掘、人工智能。胡宏银　硕士研究生，主要研究方向为数据采掘。

集。

定义2 设数据集 $E'\subseteq E$ 为不纯数据集，$X_L'=\{l_1,\cdots,l_s\}$ 为 E' 的类别集，类别为 l_i 的数据子集是 $E'_{l_i}=\{e\mid e\in E'\wedge f(e)=l_i\}$，它在 E' 中所占的比率 $p_{l_i}=\frac{|E'_{l_i}|}{|E'|}$，不纯数据集 E' 的不纯度记作：impurity(E')

$$\text{impurity}(E')=-\sum_{i=1}^{|X_L'|}p_{l_i}\log p_{l_i}$$

定义3 设当前个体是 $w=(w_1,w_2,\cdots,w_r,w_{r+1})$，对应的线性联合属性为 $\sum_{i=1}^{r}w_ia_i-w_{r+1}$，它把 E' 划分为两部分，划分后的不纯度为 impurity(w,E')：

$$\text{impurity}(w,E')=\frac{|E'_{left}|}{|E'|}\text{impurity}(E'_{left})+\frac{|E'_{right}|}{|E'|}\text{impurity}(E'_{right})$$

其中：$E'_{left}=\{e_i\mid e_i\in E'\wedge\sum_{j=1}^{r}w_jx_{ij}+w_{r+1}\leqslant 0\}$

$$E'_{right}=\{e_i\mid e_i\in E'\wedge\sum_{j=1}^{r}w_jx_{ij}+w_{r+1}>0\}$$

不纯度的减小量为 $\Delta\text{impurity}(w,E')=\text{impurity}(E')-\text{impurity}(w,E')$，它表示对应线性联合属性的判别能力，选择 $\Delta\text{impurity}(w,E')$ 为适应度函数。

适应度是衡量个体优劣的尺度，是驱动遗传算法的动力，它影响着遗传算法的效率。当前种群中，适应度大的被复制，小的被淘汰，使新群体中的个体总数与原来群体相同。被复制个体的数目由复制概率 P_r 控制，P_r 常取0.1～0.2；复制对象通常用 J. Holland 提出的轮盘方式选择。

2.2 杂交

杂交是将两个个体的某些基因相互交换，从而生成新的个体，它是产生新个体的主要手段。

定义4 设随机确定个体 w_i 和 w_j 进行杂交，杂交系数 $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_r,\alpha_{r+1})$，$0<\alpha_i<1$，$i=1,\cdots,r+1$，$w_{ik}$ 和 w_{jk} 分别为 w_i 和 w_j 杂交前的第 k 个基因，w'_{ik} 和 w'_{jk} 分别是杂交后的基因，杂交操作如下：

$$w'_{ik}=w_{jk}(1-\alpha_k)+\alpha_kw_{jk}$$

$$w'_{jk}=w_{jk}(1-\alpha_k)+\alpha_kw_{ik}$$

由定义可知，$w_{ik}+w_{jk}=w'_{ik}+w'_{jk}$，表明杂交是在对等基因间进行，且杂交前后对等基因之和相等。

被杂交个体的数目由杂交概率 P_c 控制，P_c 常取0.5～0.8；相互交换的个体随机确定；杂交的位置由杂交系数控制，杂交系数随机产生。

2.3 突变

突变是使个体的某些基因发生跳跃性变化，它是遗传算法跳出局部域，寻求全局优化解的主要手段。

定义5 设随机确定 w_i 进行突变，随机突变系数 $\beta=(\beta_1,\beta_2,\cdots,\beta_r)$，$\delta=(\delta_1,\delta_2,\cdots,\delta_r)$，$0<\beta_i<1$，$0<\delta_i<1$，$i=1,\cdots,r-1$，$w_{ik}$ 为 w_i 突变前的第 k 个基因，w'_{ik} 是突变后的基因，突变如下：

$$w'_{ik}=\begin{cases}w_{ik}-(w_{ik}-1)\delta_k & \text{当}0<\beta_k\leqslant 0.5\\ w_{ik}+(1-w_{ik})\delta_k & \text{当}0.5<\beta_k<1\end{cases}$$

当 $w'_{ik}<-1$ 时，取 $w'_{ik}=-1$；当 $w'_{ik}>1$ 时，取 $w'_{ik}=1$。被突变个体的数目由突变概率 P_m 决定，P_m 一般取0.001～0.01；突变个体随机确定；突变的位置由突变系数控制、突变系数随机产生。

2.4 算法

设当前节点对应数据集为 E'，第 i 代群体表示为 $W_i=\{w_1,w_2,\cdots,w_{PoSum}\}$，PoSum 为种群规模，第 i 代群体中最优个体为 $OptW_i$，对应适应度为 $OptF_i$，全局最优个体为 OptW、对应适应度为 OptF；gen 为代数计数器，最大代数为 MaxGen，

种群规模：个体数越多，搜索范围越广，容易获取全局最优解；但若个体数目太多，每次迭代时间过长，影响搜索效率。本文取种群规模 PoSum=100。

初始群体：选择最佳单属性作为一个初始个体，设最佳单属性形式为 $w_ia_i+w_{r+1}$，则个体中除第 i 个和第 $r-1$ 个基因外，其余基因都为0，表示为 $w_i=(0,\cdots,w_i,\cdots,w_{r+1})$；其余99个个体随机产生。

用遗传算法优化线性联合属性系数是首先把系数进行编码，然后执行遗传算法的复制、杂交、突变等操作，一步步地寻找最佳联合属性。我们称采用实数编码遗传算法的算法为 DGLC(Decimal Genetic algorithm construct Linear Combined feature)。

算法1：寻找最佳线性联合属性算法 DGLC

步骤一：建立初始种群 $W_0=\{w_1,w_2,\cdots,w_{100}\}$，计算各个个体的适应度，最优适应度为 $OptF_0$、最优个体为 $OptW_0$；设置 $OptF=OptF_0$、$OptW=OptF_0$，gen=0；设置 W_0 为当前群体。

步骤二：gen=gen+1；

处理当前种群、执行如下操作：

复制：取 $P_r=0.1$、用轮盘法选择10%的个体到 W_{gen} 中；

杂交：取 $P_c=0.7$，对70%的个体进行不重合的配对，随机产生杂交系数，执行杂交操作、产生的新个体加入到 W_{gen} 中；

变异：随机选择5个个体进行变异，变异系数随机产生，执行变异操作，产生的新个体加入到 W_{gen} 中。

步骤三：对于 W_{gen}，计算各个个体的适应度，最大的适应度为 $OptF_{gen}$，相应的个体为 $OptW_{gen}$。

如果 gen≥MaxGen，则转步骤四；

如果 gen≤MaxGen 并且 $OptF_{gen}\leqslant OptF$，则把 W_{gen} 作为当前种群、转步骤二；

如果 gen≤MaxGen 并且 $OptF_{gen}>OptF$，则设置

$OptW=OptW_{gen}$，$OptF=OptF_{gen}$，并把 W_{gen} 作为当前种群，转步骤二。

步骤四：设 $OptW=(w_1,w_2,\cdots,w_{r+1})$，则返回最佳线性联合属性 $\sum_{i=1}^{r} w_i a_i + w_{r+1}$。

图1 寻找最佳线性联合属性算法 DGLC

3 构造斜决策树

构造斜决策树采用自上而下的生长方式。我们称采用 DGLC 的算法为 ODDG(Oblique Decision tree with Decimal Genetic algorithm)。

设训练集为 E，当前节点的数据集为 E'，X_i 和 X'_i 为类别属性。

算法2：构造斜决策树算法 ODDG

步骤一：处理训练集 E；

若 E 是纯的，类别属性值为 X_i，则选择 X_i 作为叶节点(也是根节点)，转步骤五；

否则，把 E 作为当前数据集 E'。

步骤二：处理当前数据集 E'；

用算法 DGLC 选择最佳线性联合属性作为节点，根据线性联合属性的取值进行分枝，并把数据集划分为两个子数据集。

步骤三：逐个处理子数据集；

若子数据集是纯的，类别属性值为 X'_i，则选择 X'_i 作为叶节点，转步骤四；

否则，把该子数据集作为当前数据集，转步骤二。

步骤四：若子数据集处理完，转步骤五；

否则，转步骤三。

步骤五：反回决策树。

图2 构造斜决策树算法 ODDG

4 实验结果

实验的目的是检验算法 ODDG 的性能，其性能体现在决策树的规模，决策树的预测精度、决策树构造的时间、节点的可理解性和整个树的可理解性等方面，本实验比较两个主要的参数：预测精度和树的大小。预测精度指对测试集的分类精度，树的大小指叶节点的数目。

我们把 ODDG 同 OC1[3]、CART_LC[2]和 C4.5[1]三个算法进行了比较，其中 OC1、CART_LC 是斜决策树算法，C4.5是单元树算法，为了进行比较，我们设计了二进制编码的遗传算法，寻找线性联合属性的算法称为 BGLC(Binary Genetic algorithm construct Linear Combined feature)，采用 BGLC 构造决策树的算法为 ODBG(Oblique Decision tree with Binary Genetic algorithm)，具体算法从略。

从 UCI machine learning respository 选取 Satellite 和 Letter 两个数据集，前者有6435个事例，36个数值型描述属性，类别属性有6个；后者有20000个事例，16个数值型描述属性，类别属性有26个值(26个英文字母)。

用10-fold cross validation 进行实验，即把数据集分为10等份，每次取一份作为测试集，其它的作为训练集，重复10次，实验结果为10次的平均值。

算法	Satellite		Letter	
	精度	大小	精度	大小
ODDG	96.8%	3.4	75.6%	5.4
ODBG	95.5%	3.1	74.5%	6.1
OC1	95.4%	3.2	73.5%	6.3
CART_LC	93.2%	4.3	71.1%	9.2
C4.5	91.2%	11.4	70.2%	50.4

由上表可得出以下结论：

(1)多元树算法(如 ODDG、OC1等)与单元树算法(C4.5)相比，前者构造的树规模小，精度高，但是构造速度慢许多(具体时间没有列出)；

(2)基于遗传算法的斜决策树构造算法(ODDG 和 ODBG)与其它典型的斜决策树构造算法相比，两者在树的规模上差不多，但前者的预测精度略有提高；在执行时间上，前者与初始群体的选择有关，有时较长，有时比较短。

(3)基于实数编码遗传算法 ODDG 与基于二进制编码遗传算法 ODBG 相比，两者在预测精度和树的规模差不多，但前者的构造时间比后者稍短。

因此，用遗传算法优化线性联合属性的系数，进而构造决策树是一种有效的方法，但 ODDG 也有缺陷，例如，若初始群体选择不当，会出现"早熟"的可能；另外，实数编码的遗传算法与二进制编码遗传算法在实验上没有很大的区别，但前者缺乏理论上的支持，这也是我们以后的研究方向。

参考文献

1 胡宏银，朱绍文，王泉德，等. 从数据库中获取信息的几种方法的比较. 见：1999年中国智能自动化学术会议论文集. 清华大学出版社，1999. 812～817

2 Breiman L, et al. Classification and Regression Trees. Wadsworth Int. Group, 1984

3 Murthy S, Kasif S, Salzberg S. A system for induction of oblique decision trees. Journal of Artificial Intelligence Research, 1994, (2): 1～32

4 Heath D, Kasif S, Salzberg S. Induction of oblique decision trees. In: The Thirteenth Int. Joint Conf. on Artificial Intelligence. Chambery, France, 1993. 1002～1007

5 云庆夏，黄光球，王战权. 遗传算法和遗传规划——一种搜索寻优技术. 冶金工业出版社，1997. 21～32

6 Quinlan J R. C4.5: Programs for Machine Learning. Morgan Kaufmann Pub., San Mateo, CA, 1993