DOI:10.19789/j.1004-9398.2018.04.003

基于遗传算法的随机森林算法优化研究

李 东 贾郭军*
(山西师范大学数学与计算机科学学院,山西 临汾 041004)

摘 要

本文提出了一种新的基于遗传算法的随机森林的子森林选择方法,选择高质量的决策树加入初始种群生成子森林,以减小随机森林的规模并提高分类精度. 在UCI数据集上进行的实验验证了该方法的有效性.

关键词: 随机森林,决策树,分类,遗传算法.
中图分类号: TP18

0 引 言

组合分类器的基分类器的数量和多样性对其性能的影响至关重要,组合差异性小的基分类器不能有效提升组合分类器的性能[1]. 随机森林[2]组合了Bagging和随机子空间两种技术,是一种先进的决策树组合算法,为了获得更好的组合精度,需要保证单棵决策树的精度和多样性[3]. 决策树组合算法的一个主要缺点是需要大量的决策树来保证收敛性. 生成大量的决策树需要大量的内存和计算开销. 然而,森林中的树对于算法的贡献是不一样的,其中一些树可能扩大错误预测,以至于降低森林的预测性能. 因此,通过修剪一些相对有害的树得到的子森林比完整的森林有更好的性能[4].

一个森林有T棵树,那么它最多有(2^T-1)个非空子森林,从中找到最优子森林的方法是穷举法,但是当树的数量过多时,候选的子森林呈指数式增长,因此这种方法是不现实的. 为了避免穷举法的计算成本,文献[4,6,9,10]提出了贪心法,有些方法本质上是启发式的,根据单棵树的贡献排序,定义一个数,按顺序选择决策树组成子森林. 如文献[5]为AdaBoost提出了一种基于启发式的Kappa修剪技术,还有"个体错误最小化剪枝"[6],"个体贡献排序的组合修剪(EPIC)"[4]等. 这些基于启发式的方法的主要问题是需要用户定义子森林的大小,这可能得到次优的子森林. 使用爬山算法的技术是从初始森林(空或满)开始,增加或删除决策树,以提高森林的精度和多样性[6,9]. 如基于一般的爬山策略的森林修剪技术"减少错误剪枝(REP)"[6,8-9],"辅助性方法"[3],"Margin距离最小化"[3]和"定向排序"[11].

一般来说,贪心法在计算上是廉价的,但是会陷入局部最优解. 为了克服这个问题,可以使用遗传算法(GA)来提高选择最优的或接近最优的子森林的概率. 尽管这种算法是计算密集型的,但是与原始森林的大小不是指数关系[11],而且基于GA的技术不需要用户定义子森林的大小. 一个有效的子森林需要高质量的树,排除低质量的树,单独使用精度或多样性评价森林中的树是不够的,任何分类器的集成同时考虑精度和多样性才会有更好的泛化性能[5],因此子森林应该尽可能多的同时选择精度高和多样性的树. 而文献[3,12]提出的基于GA的技术是从原始的森林中随机选择决策树作为初始种群,这些树可能不是高质量的,在有限的迭代中找到最优的或接近最优的子森林比较困难. 而在GA中,使用好的初始种群通常会得到好的结果[17]. 基于此,本文提出一种基于GA的随机森林优化模型,同时选择精度高和多样性的树作为GA的初始种群,以产生一个有效的子森林. 在10个UCI机器学习的数据集上的实验结果表明,以精度高和多样性的决策树作为GA的初始种群是有效的.

收稿日期: 2017-09-28
* 通信作者

1 基于 GA 的随机森林优化算法

所提出的森林修剪(即子森林的选择)主要作用是选择高质量的树加入到 GA 的初始种群. GA 是一种进化算法[14],以一种简单的数据结构编码一个潜在的解决方案,称为染色体,通常以随机定义的染色体作为初始种群,通过交叉变异产生新的种群. 然后对染色体进行评价,好的染色体表示好的解决方法,会有更多的机会产生下一代. 最好的染色体表示最好的解决方法,作为 GA 的输出. 选择初始种群首先要解决的问题是如何选择高质量的树.

1.1 选择高质量的决策树

在随机森林中,以个体的精度和多样性确定高质量的树. 因此可以找到树集 T^A,与单棵树的平均精度是 A 的森林 $T=\{T_1,T_2,\cdots,T_z\}$ 相比,其精度更高(或相等),T^A 表示如下,其中 t_i^A 是 T_i 的精度.

$$T^A=\{T_i: t_i^A \geqslant A\},\text{其中 } A=\frac{1}{|T|}\sum_{j=1}^{|T|}t_j^A. \quad (1)$$

同样,可以找到树集 T^D,与单棵树的平均多样性值是 K 的森林相比,其多样性更强(或相等),T^D 表示如下:其中 t_i^K 是 T_i 的多样性值(Kappa).

$$T^D=\{T_i: t_i^k \leqslant K\},\text{其中 } K=\frac{1}{|T|}\sum_{j=1}^{|T|}t_j^K. \quad (2)$$

Kappa 值估计两棵树 T_i 和 T_j 之间的多样性,多于两棵树时,计算每棵树 T_i 与其余的树之间的 Kappa(k)值[7]. 组合分类器的预测(多数表决)可以视作一棵树 T_j,然后计算 T_i 和 T_j 之间的 Kappa 值,如公式(3),$\Pr(a)$ 表示 T_i 和 T_j 分类的一致性,$\Pr(e)$ 表示 T_i 和 T_j 分类的机遇一致性. K 值越低,多样性越高.

$$K=\frac{\Pr(a)-\Pr(e)}{1-\Pr(e)}. \quad (3)$$

应用公式(1)和(2),可以识别出随机森林中高质量的树,这些树的精度和多样性都比平均值高或相等.

1.2 生成高质量的子森林

选择出高质量的树后,本文为随机森林生成高质量的子森林的方法如下:

(a) Sub_A:该子森林(应用公式(1))的树的精度比原始森林的树的精度的平均值 A 高或相等.

(b) Sub_D:该子森林(应用公式(2))的树的多样性比原始森林的树的多样性的平均值 D 高或相等. 注意,公式(2)中用 K 值测量 D 值,K 值越低,D 值越高.

(c) $Sub_{A,D}$:该子森林的树的精度和多样性都比原始森林的平均值高或相等($Sub_{A,D}=Sub_A\cap Sub_D$).

(d) Sub_{All}:包含原始森林所有的树.

由 2.1 节实验表 2 可知 $Sub_{A,D}$ 中的树质量很高但数量很少. 为了增加高质量树的数量,同时保持其精度和多样性比其它树高,首先计算原始森林单棵树的精度的标准差 α 和多样性的标准差 δ. 然后生成 $Sub_{A-\alpha}$,即将原始森林中精度大于等于 $A-\alpha$ 的树加入到 Sub_A. 同样,生成 $Sub_{D+\delta}$,即将原始森林中多样性大于等于 $D+\delta$ 的树加入到 Sub_D. 然后生成 $Sub_{A-\alpha,D+\delta}=Sub_{A-\alpha}\cap Sub_{D+\delta}$,进而生成 $Sub_{A-2\alpha,D+2\delta}=Sub_{A-2\alpha}\cap Sub_{D+2\delta}$ 作为 $Sub_{A,D}$ 的一个扩展. 这个扩展如图 1 所示.

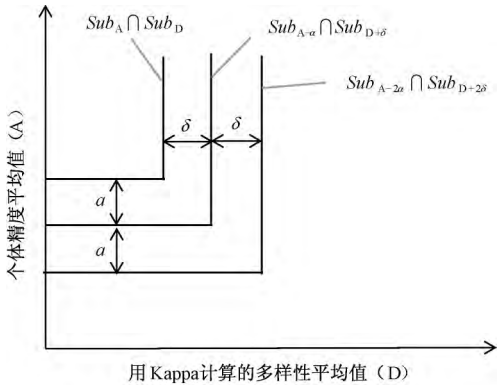

图 1 子森林扩展

由 2.1 节实验结果可知生成的子森林的精度都较高,但树的数量差距较大,因此使用遗传算法从这些子森林中选择较好的树加入初始种群,以生成最优子森林.

1.3 初始种群的选择

种群是一个染色体组,首先对染色体编码. 用 T 个二进制数编码染色体,第 i 位表示第 i 棵树. 每一位有两个值,1 表示该树被选择,0 表示该树没有被选择. 例如,染色体 $Cr=0100010110$ 表示 10 棵树中,第 2,6,8,9 棵树被选择作为子森林. 本文中,一个森林生成 100 棵树,一个染色体代表一个子森林,设置森林中树的数量为 $T=100$.

从 2.1 节实验表 3 的平均值看,随机森林算法的 $Sub_{A-\alpha,D+\delta}$ 优于 Sub_A 和 Sub_D,即从高质量的树中产生的子森林优于其它相近大小的子森林,因此为 GA 选择高质量的树. 文献[8,12]子森林的 M 棵树是从 T 中随机选择的,然而,随机选择树的子森林无法保证有足够多的高质量的树,而 GA 算法的交叉

和变异也是随机的,很难在一个合理次数的迭代中生成最优或接近最优的子森林. 另一方面,仅有高质量的树不能生成最优的子森林,如 $Sub_{A\cap D}$ 表现不好. 由于 $Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 表现相对较好,因此高质量的树会优先缩小 GA 的搜索空间. 因此,本文在编码染色体时既选择高质量的树,也随机选择树.

本文使用文献[12]的方法,将 20 条染色体作为一个种群. 在种群中,奇数的染色体(共 10 条)是相对高质量的树编码的,偶数的染色体(共 10 条)是随机选择的树编码的[8,12]. 高质量的树编码染色体方法如下:

为了对奇数染色体中相对高质量的树排序,采用分层抽样代替随机抽样. 分层抽样是概率抽样,首先将数据集分成若干个互斥的、均匀的层,然后按比例或不按比例抽取样本[15]. 本文定义 3 层:

(1) 第 1 层 S_1: $Sub_{A,D}$ 的树;

(2) 第 2 层 S_2: $Sub_{A-\alpha,D+\delta}\backslash Sub_{A,D}$ 的树;

(3) 第 3 层 S_3: $Sub_{A-2\alpha,D+2\delta}\backslash Sub_{A-\alpha,D+\delta}$ 的树.

随机森林服从钟型分布,从 2.1 节表 3 平均值看 S_1,S_2 和 S_3 总共覆盖 95.18% 的树. 其余 4.82% 的低质量的树被排除. 定义分层后,M 值在 1 到 100 的范围内随机选择. 如果 $M > |S_1|+|S_2|+|S_3|$,则 M 设置为 $|S_1|+|S_2|+|S_3|$. 从 S_1,S_2 和 S_3 中使用不等比例分层抽样(DSS)抽取 M 棵树. 本文的 DSS 优先权为 $S_1 > S_2 > S_3$. 例如: $M = |S_1|$,染色体的所有的树从 S_1 中选择; 当 $M \leqslant |S_1|$ 时,从 S_1 中随机选择 M 棵树; 当 $M > |S_1|$ 且 $M < |S_1|+|S_2|$ 时,首先选择 S_1 中所有的树,剩下 $M - S_1$ 棵树从 S_2 中随机选择; 当 $M = |S_1|+|S_2|$ 时,选择 $S_1 \cup S_2$ 中所有的树,以此类推.

编码偶数染色体时,M 值是从 1 到 100 随机选择. 这 M 棵树用文献[8,12]的方法从 T 中随机选择,T 中可能包含 4.82% 的低质量的树.

当这 20 条染色体生成后,设置其为初始种群 P_{Curr}. P_{Curr} 的后代中最好的染色体替换当前最好的染色体 Cr_{SFBest}.

1.4　交叉和变异

交叉是 GA 算法中种群进化的一个重要组成部分[14,17]. 一般的,交叉在染色体对(双亲)上进行,交换片段后形成后代. 本文中,选择 P_{Curr} 中最好的染色体 Cr_b 作为第一对儿染色体. 选择第二对染色体时使用轮盘赌选择方法[16-17],轮盘赌具有随机性,染色体 $Cr_r(\neq Cr_b)$ 被选择的概率为 $p(Cr_r)=\dfrac{EA(Cr_r)}{\sum_{i=1}^{|P_{Curr}|}EA(Cr_i)}$(其中 $EA(Cr_r)$ 是染色体即子森林 Cr_r 的组合精度),这保证好的染色体比差的染色体有更大的概率被选择,已经选择的染色体对不会被重复选择.

染色体配对后,交叉点的操作如图 2. 每对染色体的交叉点是从 1 到 T 随机选择的,T 是完整森林的树数,是染色体的全长. 交叉是染色体对的左右基因交换. 交叉后,子代染色体对与亲代染色体对的基因数相同.

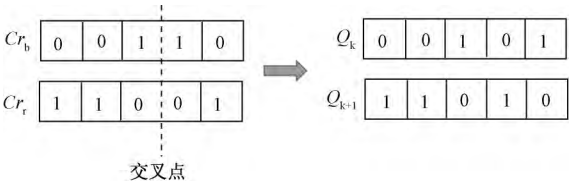

图 2　交叉

变异具有一定的随机性,对子代染色体从 1 到 T 中随机选择一个位,如果随机选择的位是 0,将它转换为 1,如果是 1,则转换为 0.

1.5　精英选择

交叉变异后,P_{Curr} 发生了变化,为新的 Cr_{SFBest} 进行精英选择[17]. 在迭代开始时,P_{Curr} 所有的染色体都复制到 P_{TempCurr}. P_{Curr} 中最好的染色体保存到 Cr_{CurrBest}. 然后第一次交叉发生在 P_{TempCurr} 以生成 P_{Mod}.

接下来,精英选择比较 Cr_{ModBest}(P_{Mod} 中最好的染色体)和 Cr_{SFBest},如果 Cr_{ModBest} 优于 Cr_{SFBest},那么用 Cr_{ModBest} 取代 Cr_{SFBest}. 如果 Cr_{CurrBest} 中的染色体优于 P_{Mod} 中最差的染色体,那么用 Cr_{CurrBest} 中的染色体取代 P_{Mod} 中最差的染色体,然后在 P_{Mod} 中重新计算得出 Cr_{CurrBest}. 这样,一个更好的 Cr_{CurrBest} 总会保留在 P_{Mod} 中,这保证选出最优的染色体. P_{Mod} 进一步变异,再用同样的方式进行精英选择.

1.6　下一次迭代的染色体选择

每次迭代结束,检查遗传操作所做的修改是否不利,即 P_{Mod} 种群中的大部分染色体在迭代结束时(交叉,变异和精英选择后)的 P_{Curr} 染色体是差的. 这是下一次迭代的输入种群,这种不利的修改表明 P_{Mod} 不如 P_{Curr}. 注意,交叉和变异应用在 P_{TempCurr},因此 P_{Curr} 在迭代过程中保持不变. 如果 P_{Mod} 不如 P_{Curr},那么 P_{Mod} 成为 P_{Curr} 做下一次迭代时,种群质量下降可能性很大,导致在解空间错误的方向搜索.

为了防止质量下降的情况,在每次迭代结束时,创建一个染色体池 P_{Pool},加入 P_{Curr} 和 P_{Mod} 的一些染色体,因此 P_{Pool} 由 40 条染色体组成. 然后用轮盘赌方法从 P_{Pool} 中选出20 条染色体,这20 条染色体组成新的 P_{Curr} 做下一次迭代,这使好的染色体能进行下一次迭代.

1.7 染色体的校正

GA 能够避开局部最优解,并且能够在迭代的交叉和变异下在大的解空间内搜索,但是,潜在的解决方案是随机的,因此,可以在最优/接近最优的解的方向上微调. 本文使用文献[12]中在迭代结束时的 Cr_{SFBest} 上应用的局部搜索方法的组合搜索法(CSO)校正 Cr_{SFBest},方法如下:

第 1 步,Cr_{SFBest} 上 1 依次变异为 0(即每棵树都被排除在当前最好的子森林之外),每一位从 1 变为 0 之后,对染色体的 EA 进行检查,如果 EA 值没有增加,0 重新变为原来的 1,否则,维持这种变异. 第 2 步,Cr_{SFBest} 上 0 依次变异为 1(即每一棵树都包含在当前最好的子森林中),每一位从 0 变异为 1 后,对染色体的 EA 进行检查,如果 EA 值没有增加,1 重新变为原来的 0,否则,维持这种变异.

2 实验验证

本文实验分为两部分,验证所选子森林的质量和比较不同算法的子森林的性能. 实验数据集为 10 个UCI 数据集(如表 1). 实验中,移除了缺失值数据(表 1 没有显示缺失值数据)和标识符属性(如每个数据集的 Transaction_ID). 实验环境为: Intel(R) 3.4 GHz 处理器,8 GB 内存(RAM),64 位 Windows7 旗舰操作系统.

表 1 实验数据集

数据集(DS)	样本数	非类属性	类属性
Chess (CHS)	3 196	36	2
Credit Approval (CA)	653	15	2
Dermatology (DER)	358	34	6
Hepatitis (HEP)	80	19	2
Libras Movement (LM)	360	90	15
Liver Disorder (LD)	345	06	2
Pima Indians Diabetes (PID)	768	08	2
Statlog Heart (SH)	270	13	2
Statlog Vehicle (SV)	846	18	4
Thyroid-New (TN)	215	05	3

2.1 验证子森林的质量

从大的训练集生成的分类器,其组合分类器和修剪后的组合分类器性能都更好[11],因此对每个数据集使用十折交叉验证(10-CV)[13]. 10-CV 是将数据集随机分成 10 份,轮流将 1 份数据作为测试数据(袋外数据),其余 9 份数据作为训练数据,因此得到 10 个训练集和 10 个相应的测试集.

实验中使用基尼系数作为随机森林的分裂准则. 所有树节点的分裂属性的基尼系数均设置为 0.01. 每个叶节点最少包含 2 条数据,不会进行剪枝,以使结果不受剪枝方法的影响. 使用决策树分类结果进行投票,票数最多的类为森林的分类结果. 为每个数据集的随机森林生成 100 棵树,这个数是足够大的,以确保组合分类器的收敛性[7]. 每个训练集生成 100 棵树,总共能生成 1 000 棵树,然后用相应的测试集评估它们的性能. 以两个重要的性能指标组合的精度(EA 百分比)和组合的大小(ES 树的数量)评估子森林性能. 实验结果如表 2 所示.

表 2 随机森林的子森林 Sub_A、Sub_D、$Sub_{A,D}$和 Sub_{All}的 EA 和 ES

数据集	EA				ES			
	Sub_A	Sub_D	$Sub_{A,D}$	Sub_{All}	Sub_A	Sub_D	$Sub_{A,D}$	Sub_{All}
CHS	94.84	83.97	82.16	95.22	72.40	42.90	15.30	100.00
CA	86.68	81.16	83.76	86.07	76.80	40.90	18.50	100.00
DER	87.82	62.03	73.37	86.96	64.40	37.20	4.50	100.00
HEP	86.25	85.00	81.25	86.25	71.70	43.40	17.80	100.00
LM	74.72	75.00	51.11	76.11	50.20	49.50	2.40	100.00
LD	68.40	68.36	66.63	71.48	62.30	42.70	20.20	100.00
PID	76.22	75.83	75.38	75.95	60.20	41.90	22.50	100.00
SH	82.96	82.97	80.37	82.96	51.40	47.00	10.70	100.00
SV	74.27	74.02	73.05	74.14	59.70	42.90	13.30	100.00
TN	95.04	95.99	93.90	94.56	60.50	38.70	8.30	100.00
平均值	82.72	78.43	76.10	82.97	62.96	42.71	13.35	100

从表 2 结果可以看出,随机森林生成的子森林(Sub_A,Sub_D,$Sub_{A,D}$)的 EA 值比原始森林低. 值得注意的是,$Sub_{A,D}$ 是其中最差的,这是因为树的数量很少而质量很高,这对于大多数数据集不能保证组

合分类器的收敛性. 因此将子森林 $Sub_{A,D}$ 扩展为 $Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 以增加高质量树的数量.

表 3 列出了随机森算法的子森林在 10 个数据集上的 EA 和 ES 的平均值. 可以看出,随着 ES 的增加 EA 值并没有逐渐增加,其中 $Sub_{A-\alpha,D+\delta}$ 的 EA 值是最大的,需要注意的是, $Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 都优于 Sub_A,Sub_D 和 Sub_{All}. 这表明单棵树的精度和多样性有助于提高 EA. 此外,类似文献 [4,6,9-11] 的研究结果, $Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 的 EA 值优于完整的森林 Sub_{All}, 这是因为它们排除了一些树,这些树的精度和多样性比大多数的树低.

表 3　比较子森林 Sub_A、Sub_D、$Sub_{A,D}$、$Sub_{A-\alpha,D+\delta}$、$Sub_{A-2\alpha,D+2\delta}$和 Sub_{All}的性能

子森林	Sub_A	Sub_D	$Sub_{A,D}$	$Sub_{A-\alpha,D+\delta}$	$Sub_{A-2\alpha,D+2\delta}$	Sub_{All}
EA 平均值	82.72	78.43	76.10	84.06	83.91	82.97
ES 平均值	62.96	42.71	13.55	78.20	95.18	100.00

从表 3 还可以看出随机森林的 ES 的增加服从钟型分布, $Sub_{A,D}$ 包含 13.55% 树,$Sub_{A-\alpha,D+\delta}$ 包含 78.20% 的树,$Sub_{A-2\alpha,D+2\delta}$ 包含 95.18% 的树. 这表示对于随机森林的大多数树,高质量的树和低质量的树是有限的,其中高质量的树占 13.55%,低质量的树占 4.82%. 尽管 $Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 的 EA 值非常接近,但树的数量差距很大. 因此,接下来使用 GA 寻找最优子森林,即从子森林 Sub_A、Sub_D、$Sub_{A,D}$、$Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 中选择较好的树作为初始种群.

2.2　比较不同算法的子森林的性能

对比的子森林算法有: 基于个体精度的排序树(Sub_{IA}) [8],基于个体多样性的排序树(Sub_{ID}) [8], EPIC(Sub_{IC}) [4],基于遗传算法的剪枝方法 HGA (Sub_{HGA}) [12].

Sub_{HGA} 在下一次迭代中没有使用染色体选择,但是,从文献 [18] 可以看出,使用染色体选择可以提高整体性能,因此,本文中 Sub_{HGA} 和 Sub_{PGA} 均使用这个操作,以对比 Sub_{HGA} 和 Sub_{PGA} 的性能. 此外,本文中 Sub_{HGA} 和 Sub_{PGA} 均使用染色体校正.

同文献 [4] 一样, Sub_{IA},Sub_{ID} 和 Sub_{IC} 从 100 棵树的森林中分别生成 40,60 和 80 棵树的子森林. 随机森林的子森林 Sub_{IA}、Sub_{ID} 和 Sub_{IC} 的 EA 值如表 4.

表 4　随机森林的子森林 Sub_{IA}、Sub_{ID}和 Sub_{IC}的 EA 值

数据集	Sub_{IA}^{40}	Sub_{IA}^{60}	Sub_{IA}^{80}	Sub_{ID}^{40}	Sub_{ID}^{60}	Sub_{ID}^{80}	Sub_{IC}^{40}	Sub_{IC}^{60}	Sub_{IC}^{80}
CHS	94.72	94.65	95.12	80.77	90.93	95.53	92.31	95.69	96.00
CA	86.37	86.83	86.22	81.31	84.07	86.70	85.31	86.22	86.37
DER	90.97	88.96	87.54	65.01	80.94	86.84	89.30	87.87	86.16
HEP	85.00	85.00	86.25	85.00	86.25	87.50	85.00	87.50	86.25
LM	75.83	74.72	76.11	75.00	73.89	75.83	73.61	74.45	76.67
LD	69.42	68.91	70.93	68.87	70.67	71.77	70.34	71.89	72.40
PID	75.72	76.75	75.85	76.08	75.20	75.85	76.07	76.47	76.21
SH	81.48	84.45	83.70	82.22	83.34	82.59	82.22	82.59	83.33
SV	74.63	74.27	74.15	74.14	73.56	72.96	74.84	73.64	74.02
TN	95.51	95.04	94.56	95.99	95.04	94.56	94.56	94.56	94.56
平均值	82.97	82.96	83.04	78.44	81.39	83.01	82.36	83.09	83.20

从表 4 可以看出,数据集(问题) 不是独立于最优子森林的大小,即对一个数据集,增加一些树,EA 值可能会增加,而对于另一个数据集会减小. 从 EA 平均值看,RF 的 Sub_{IC}^{80} 性能最好. 由于贪心算法性能不稳定,进一步建立基于 GA 的算法,表 5 比较了基于 GA 的子森林算法 Sub_{HGA} 和 Sub_{PGA} 的性能.

从表 5 可以看出,对比 10 个数据集上的随机森林的子森林的 EA 值,在 8 个数据集上, Sub_{PGA} 优于 Sub_{HGA},同时,在 4 个数据集上,Sub_{PGA} 树数小于 Sub_{HGA}. 从 EA 和 ES 平均值看, Sub_{PGA} 优于 Sub_{HGA}. 表 5 结果表明了基于 GA 的子森林选择算法选择高质量的树作为初始种群有利于提高子森林的性能.

表 6 对比了随机森林的所有子森林的 EA 和 ES 的平均值. 结果表明,与其他子森林相比, Sub_{PGA} 的 EA(84.21) 最高,有相对较低的 ES,因此 Sub_{PGA} 性能更好. Sub_{PGA} 不仅增加了 RF 的 EA,并且很大程度上缩小了森林的大小. 从平均值看, Sub_{PGA} 对 RF 修剪了 55.71 棵树,仅包含 44.23 棵树. 随机森林的计算开销主要是先生成森林,然后通过森林预测数据. 通过 GA 从完整的森林生成 Sub_{PGA} 过程中增加了一些计算量,但是 Sub_{PGA} 约占原始森林 50% 的大小,预测一条数据少 50% 的投票开销,因此森林尺寸的

表 5 比较随机森林的子森林 Sub_{HGA} 和 Sub_{PGA} 的性能

数据集	EA		ES	
	Sub_{HGA}	Sub_{PGA}	Sub_{HGA}	Sub_{PGA}
CHS	97.09	97.56	42.00	42.50
CA	86.60	87.30	47.20	48.20
DER	93.25	94.06	35.10	40.40
HEP	87.50	86.25	30.70	37.20
LM	75.62	75.83	61.60	47.40
LD	69.24	71.52	46.50	51.90
PID	74.78	76.20	46.30	49.20
SH	81.85	83.70	41.40	32.50
SV	73.48	74.25	65.10	61.30
TN	95.42	95.42	39.60	32.30
平均值	83.49	84.21	45.55	44.29

表 6 对比随机森林所有子森林的性能

子森林	EA 平均值	ES 平均值	子森林	EA 平均值	ES 平均值
Sub_A	82.72	62.96	Sub_{ID}^{60}	81.39	60.00
Sub_D	78.43	42.71	Sub_{ID}^{80}	83.01	80.00
$Sub_{A,D}$	76.10	13.55	Sub_{IC}^{40}	82.36	40.00
$Sub_{A-\alpha,D+\delta}$	84.06	78.20	Sub_{IC}^{60}	83.09	60.00
$Sub_{A-2\alpha,D+2\delta}$	83.91	95.18	Sub_{IC}^{80}	83.20	80.00
Sub_{IA}^{40}	82.97	40.00	Sub_{HGA}	83.49	45.55
Sub_{IA}^{60}	82.96	60.00	Sub_{PGA}(提出的)	84.21	44.29
Sub_{IA}^{80}	83.04	80.00	Sub_{All}	82.97	100.00
Sub_{ID}^{40}	78.44	40.00			

减小能够减少存储和计算开销.

表 6 表明,与 Sub_{PGA} 相比,$Sub_{A-\alpha,D+\delta}$ 和 $Sub_{A-2\alpha,D+2\delta}$ 的树更多,但是精度没有 Sub_{PGA} 高. 因此,只增加大量的树不能得到很好的组合精度,需要恰当数量的树. Sub_{PGA} 减少了大量的树,但是精度没有大幅下降,而且获得的子森林不需要用户自定义树数. 从表 6 可以看出,Sub_{IA}^{40},Sub_{ID}^{40} 和 Sub_{IC}^{40} 生成的树数与 Sub_{PGA} 相近,精度接近 Sub_{PGA},但是这些子森林的树数是用户定义的,算法不能自动找到. 这些子森林选择技术的精度没有 Sub_{PGA} 高,即使使用 80 棵树,精度仍低于 45 棵树的 Sub_{PGA}. 很明显,如表 6 所示,Sub_{PGA} 保持最佳的预测精度的同时很大程度上减小了森林的规模,精度甚至比完整的森林高.

所提出的 Sub_{PGA} 优于贪心算法和基于遗传算法的子森林选择技术 Sub_{HGA}. 很明显 Sub_{PGA} 和 Sub_{HGA} 的主要区别是初始种群的选择. 因此本文的实验验证了提出的初始种群选择方法的有效性.

3 结 论

本文提出了一种从一个大的森林中筛选一个小的子森林的算法. 森林中的许多树对于提高森林的整体预测往往不是很有用. 此外,如果森林中的树数量巨大,预测的开销会非常大,尤其预测的数据非常多时. 因此,为了减少计算开销,提高组合精度,找到一个有效的子森林非常重要. 精度和多样性更高的树组成的子森林精度高规模小,但是得到的最佳的子森林需要恰当数量的树. 最佳子森林的树数会因为数据集的不同显著变化,对于用户/数据集来说,提前猜测是困难的. 因此,提出了一种基于遗传算法的方法自动识别树数以得到最优的或接近最优的子森林.

本文提出的 Sub_{PGA} 在基于 GA 算法上使用本文的初始种群选择方法,以较少的树获得比完整森林更高的精度. 从统计结果可看出效果明显,优于现有的一些技术,因此,提出的方法具有有效性. 未来的工作包括进一步调整遗传操作如 Sub_{PGA} 的交叉,变异和精英选择,在更多的数据集上评估有效性.

参 考 文 献

[1] Shipp C A, Kuncheva L I. Relationships between combination methods and measures of diversity in combining classifiers [J]. Information Fusion, 2002, 3(2): 135-148.

[2] Breiman L. Random forests [J]. Machine Learning, 2001, 45(1): 5-32.

[3] Polikar, Robi. Ensemble based systems in decision making [J]. IEEE Circuits & Systems Magazine, 2006, 6(3): 21-45.

[4] Lu Z, Wu X, Zhu X, et al. Ensemble pruning via individual contribution ordering [C]. Programme Technical Committee, 2010: 871-880.

[5] Tang E K, Suganthan N, Yao X. An analysis of diversity measures [J]. Machine Learning, 2006, 65(1): 247-271.

[6] Margineantu D D, Dietterich T G. Pruning adaptive boosting [C]. Fourteenth International Conference on Machine Learning, 1997: 211-218.

[7] Amasyali M F, Ersoy O K. Classifier ensembles with the extended space forest [J]. IEEE Educational Activities Department, 2014, 26 (3): 549 – 562.
[8] Ruta D, Gabrys B. Classifier selection for majority voting [J]. Information Fusion, 2005, 6 (1): 63 – 81.
[9] Martínez-Muñoz G, Suárez A. Aggregation ordering in bagging [C]. Munoz, 2004: 258—263.
[10] Partalas I, Tsoumakas G, Vlahavas I. Focused ensemble selection: a diversity-based method for greedy ensemble selection [C]. Conference on Ecai: European Conference on Artificial Intelligence. 2008: 117 – 121.
[11] Martinezmuoz G, Hernandezlobato D, Suarez A. An analysis of ensemble pruning techniques based on ordered aggregation [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2009, 31 (2): 245 – 59.
[12] Kim Y W, Oh I S. Classifier ensemble selection using hybrid genetic algorithms [J]. Pattern Recognition Letters, 2008, 29 (6): 796 – 802.
[13] Arlot S, Celisse A. A survey of cross-validation procedures for model selection [J]. Statistics Surveys, 2010, 4 (2010): 40 – 79.
[14] Whitley D. A genetic algorithm tutorial [J]. Statistics & Computing, 1994, 4 (2): 65 – 85.
[15] Daniel J. Sampling essentials: practical guidelines for making sampling choices [M]. Sage Publications, 2011.
[16] Liu Y, Wu X, Shen Y. Automatic clustering using genetic algorithms [J]. Applied Mathematics & Computation, 2011, 218 (4): 1267 – 1279.
[17] Rahman M A, Islam M Z. A hybrid clustering technique combining a novel genetic algorithm with K-Means [J]. Knowledge-Based Systems, 2014, 71 (71): 345 – 365.
[18] Beg A H, Islam M Z. Genetic algorithm with novel crossover, selection and health check for clustering [J]. European Symposium on Artificial Neural Networks, 2016: 575 – 580.

Research on Optimizing Random Forest Algorithm Based on Genetic Algorithm

Li Dong　Jia Guojun

(School of Mathematics & Computer Science, Shanxi Normal University, Linfen Shanxi　041004)

Abstract

This paper proposes a new subforest of random forest selection method based on genetic algorithm. In this method, high quality decision trees are selected to add into initial population to generate subforest to reduce the size of random forest and improve classification accuracy. Finally, experiments on UCI data sets verify the effectiveness of the proposed method.

Key words: random forest, decision tree, classification, genetic algorithm.