基于out-of-bag样本的随机森林算法的超参数估计

李 毓[1], 张春霞[2]

(1. 信阳师范学院经济与管理学院, 河南 信阳 464000;
2. 西安交通大学理学院统计金融系, 陕西 西安 710049)

摘要: 随机森林是一种有效的分类树集成算法,但为了使它具有较高的预测精度,要采用某种方法确定其超参数的最优值. 在不额外增加计算复杂性的前提下, 提出了一种基于out-of-bag样本估计其超参数取值的方法. 仿真试验的结果表明, 利用文中提出的方法所选取的超参数在多数情况下都能使随机森林算法的分类效果达到最优.

关键词: 集成学习; 随机森林; 泛化能力; Bootstrap 样本; out-of-bag样本; 交叉确认法
中图分类号: O212; TP181 **文献标识码**: A **文章编号**: 1000－5781(2011)04－0566－07

Estimation of the hyper-parameter in random forest based on out-of-bag sample

LI Yu[1], **ZHANG Chun-xia**[2]

(1. School of Economics and Management, Xinyang Normal University, Xinyang 464000, China;
2. Department of Statistics and Finance, Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Random forest (RF) is an effective decision tree ensemble method. In order to achieve its best performance, however, the optimal value of the hyper-parameter in RF needs to be estimated by an appropriate method. Under the condition that the computational cost is not additionally consumed, this paper proposes a new approach to estimate the hyper-parameter based on the out-of-bag sample. The experiments conducted by some UCI real-world data sets show that RF with the hyper-parameter estimated by the proposed method performs best in most cases.

Key words: ensemble learning; random forest; generalization capability; bootstrap sample; out-of-bag sample; cross-validation method

1 引 言

集成学习是模式识别研究领域中的一种新兴方法, 它采用多个学习机来解决同一问题, 由于它能显著提高一个学习系统的泛化能力, 对其理论和算法的研究近年来一直是机器学习领域的热点问题之一. 目前, 各种集成学习算法正在被广泛应用于解决机械故障诊断、图像处理、文本分类、企业信用评估等众多实际问题[1–3].

在构建具有较强泛化能力的集成学习机时, 基学习机的准确性(accuracy)和它们之间的多样性(diversity)是

收稿日期: 2010－07－06; 修订日期: 2011－05－09.
基金项目: 教育部人文与社会科学基金资助项目(09YJA790174); 河南省软科学基金资助项目(102400450126); 教育部博士学科点专项科研基金(20100201120048).

两个关键因素[4], 即基学习机的预测结果应尽可能地精确且不一致. 一般地, 准确性和多样性之间有一个折中, 不同的集成学习算法[5]主要是在生成基学习机时如何更好地达到这两者之间的折中存在差异.

随机森林[6](random forest, RF)算法是Breiman于2001年提出的一种简单且有效的集成学习技术. 目前, 随机森林算法已被成功应用于工程、生物医学和信息科学等众多领域中. 例如, 贾富仓等[7]将随机森林算法用于多谱磁共振图像的分割, 袁敏等[8]利用随机森林算法对膜蛋白的类型进行预测, 庄进发等[9]对随机森林算法进行了改进, 并用其对机械的故障进行诊断, Genuer 等[10]则用其进行变量选择, 以找到分类或回归问题中最重要的解释变量.

RF的主要思想是从给定的数据集中随机抽取Bootstrap样本[11], 利用每个Bootstrap样本构建一个分类树[12]. 其中在构建分类树时, 对于每个分裂结点, 随机森林算法首先从特征集中随机抽取K(算法的超参数)个特征, 再根据使信息增益比达到最大的准则从中选取一个最优特征作为分裂变量. 根据统计学习理论, Bootstrap样本可以被看作是与原数据集来自共同的总体分布, 因而可以保证每个分类树的准确性; 而抽取Bootstrap样本的随机性、分类树的不稳定性以及在每个分裂结点处通过随机抽取K个特征所加入的随机性则可以保证所生成的分类树之间的多样性.

Breiman[6]曾指出随机森林算法中超参数K的取值对最终构建的集成学习机的性能有较大影响, 并建议将K的值取为1或$\lfloor \log_2(p)+1 \rfloor$, p是事先给定的数据集中包含的特征个数, $\lfloor A \rfloor$表示小于A的最大整数. 随后, 研究者们在应用过程中, 大多都是采用启发式的方法对随机森林算法中的超参数K进行事先指定. 在文献[7−8,10]中, 令$K=\lfloor \sqrt{p} \rfloor$ 进行了试验, Panov等[13]则对超参数K采用$\lfloor \log_2(p)+1 \rfloor$和$\lfloor \sqrt{p} \rfloor$进行了试验. 文献[14]对随机森林算法中的超参数对其性能的影响进行了研究并指出, 超参数K的最优取值一般依赖于具体的问题, 且利用传统的启发式方法所选取的K 并不能使该算法的预测性能达到最优. 除了启发式方法之外, 还可以用验证集或者交叉确认法对K的最优取值进行选择, 但前者需要额外的数据, 而后者则会大大增加计算费用.

本文通过对随机森林算法中超参数K的最优取值方法进行研究, 提出了一种基于out-of-bag样本[15]的估计方法, 该方法能克服采用验证集或交叉确认法选取值的缺点, 并能选取到K 的近似最优值. 仿真试验的结果表明, 利用文中提出的方法所选取的超参数在多数情况下都能使随机森林的预测性能达到最优效果.

2　随机森林算法

为了叙述方便起见, 在此先引入一些记号. 假定$\mathscr{L}=\{(x_i,y_i)\}_{i=1}^N$为给定的一个训练数据集, 对于训练个体$(x_i,y_i)$, 其输入特征$x_i=(x_{i1},x_{i2},\cdots,x_{ip})\in\mathbb{R}^p$, 类标签$y_i\in\boldsymbol{\phi}_i=\{1,2,\cdots,J\}$ (J为类的总数), 利用$S=\{X_1,X_2,\cdots,X_p\}$表示p个特征组成的特征集. 随机森林算法的主要步骤如下.

- **输入**: 训练集$\mathscr{L}$; 分类树算法$\mathscr{C}$; 分类树总数B; 超参数K.
- **训练阶段** For $b=1,2,\cdots,B$
  - 从训练集$\mathscr{L}$中进行N次有放回随机抽样, 得到一个新的训练数据集(即Bootstrap样本)$\mathscr{L}_b=\{x_i^{(b)},y_i^{(b)})\}_{i=1}^N$;
  - 将$\mathscr{L}_b$作为分类树算法$\mathscr{C}$的输入训练出分类树C_b. 其中, 在分类树的每个分裂结点处, 先从p个特征中随机选取K个, 再通过使信息增益比达到最大的原则选择一个最优的特征作为分裂变量.

  EndFor
- **预测阶段**

  对于一个新的数据点x, 采用简单多数投票的合并准则预测其类标签为

$$C^*(x)=\arg\max_{1\leqslant j\leqslant J}\sum_{b=1}^{B}I(C_b(x)=j) \tag{1}$$

其中$I(A)$是取值1或0示性函数, 即当A为真时, 取值为1; 否则取值为0.

值得指出的是, 随机森林算法与Bagging算法[16−17]的唯一区别在于利用每个Bootstrap样本构建分类树的过程. 在分类树的每个分裂结点处, Bagging算法是从所有的p个特征中选择最优的分裂变量, 而随机森林算法则通过超参数K在其中加入了额外的随机性, 进而增加了分类树之间的多样性. 当$K = p$时, 这两个算法是相同的.

3 基于out-of-bag样本的随机森林算法的超参数估计

为了使随机森林算法的预测效果达到最优, 其超参数K的值需要采用某种方法进行确定. 对于该问题, 尽管目前已经有一些解决办法[6,13−14], 但这些方法大多是通过启发式的方法对K的值进行选取, 试验结果表明它们最终并不能使得随机森林算法的泛化能力达到最优. 此外, 尽管采用交叉确认法可以选取的到K的最优值, 但其计算费用往往较大, 这对于大规模的数据集更是一个难以有效解决的问题.

注意到在Bootstrap随机抽样中, Bootstrap样本只能包含原数据集中大约63.2%的个体, Breiman [15]将那些没有在Bootstrap样本中出现的个体组成的集合称为out-of-bag样本, 并指出可以用它来改善类密度估计等. 目前, out-of-bag样本已经在集成学习的研究中得到了广泛应用, 如Bylander [18]利用out-of-bag样本来估计Bagging 算法在二分类数据集上的泛化误差; Hothorn等 [19]将out-of-bag样本用于在构建每个基分类器时, 计算额外的输入特征; Martínez-Muñoz 等 [20]则利用out-of-bag样本来估计在Bagging算法中, Bootstrap样本与原训练集的样本容量的最优比例, 以使得所构建的Bagging集成达到最优的分类效果.

在此提出一种利用out-of-bag样本估计随机森林算法中超参数K最优取值的方法, 其优点是不需要额外地增加计算量. 该方法的具体步骤描述如下.

输入: 训练集$\mathscr{L}$; 分类树算法$\mathscr{C}$; 分类树总数B.

输出: 超参数K的最优取值K^*.

算法的主要步骤如下.

- For $K = 1, 2, \cdots, p$
  - 令$\boldsymbol{D}$表示元素全为0的$N \times B$阶矩阵, 即$\boldsymbol{D}_{N\times B} = \mathbf{0}$;
  - For $b = 1, 2, \cdots, B$
    1. 从训练集$\mathscr{L}$中随机抽取Bootstrap样本$\mathscr{L}_b = \{(x_i^{(b)}, y_i^{(b)})\}_{i=1}^N$;
    2. 对$\mathscr{L}$中的每个个体$(x_i, y_i)(i = 1, 2, \cdots, N)$,令
    $$D_{ib} = \begin{cases} 1, \ (x_i, y_i) \notin \mathscr{L}_b; \\ 0, \ (x_i, y_i) \in \mathscr{L}_b; \end{cases} \tag{2}$$
    3. 将相应的K值和Bootstrap样本$\mathscr{L}_b$作为分类树算法$\mathscr{C}$的输入, 训练出分类树C_b.
  - EndFor
  - 对于个体$(x_i, y_i)(i = 1, 2, \cdots, N)$, 寻找不包含它的Bootstrap样本, 令$V_i$表示这些Bootstrap样本的编号组成的集合, 即$V_i = \{b : D_{ib} = 1\}$. 然后, 利用分类树集合$C_{b\in V_i}$对x_i进行预测, 令分类树集成C^*对其预测的类标签为$y_i^* = \arg\max\limits_{1\leqslant j\leqslant J} \sum\limits_{b\in V_i} I(C_b(x_i) = j)$;
  - 估计随机森林算法的泛化误差为$\varepsilon(K) = (1/N)\sum\limits_{i=1}^{N} I(y_i \neq y_i^*)$.
- EndFor
- 估计超参数K的最优取值$K^* = \arg\min\limits_{1\leqslant K\leqslant p} \varepsilon(K)$.

上述方法的主要思想是对K的每个可能取值, 基于out-of-bag样本估计其对应的泛化误差, 则最小泛化误差对应的K值即是最终要选取的最优值. 在利用out-of-bag估计泛化误差的过程中, 对训练集$\mathscr{L}$中的每个

个体(x_i, y_i), 只采用基于不包含它的Bootstrap样本所训练的分类树$C_{b\in V_i}$对其进行预测, 再利用简单多数投票的方法估计其类标签y_i^*. 值得指出的是, 基于out-of-bag样本估计出的泛化误差$\varepsilon(K)$是无偏的, 尽管它也是基于训练集$\mathscr{L}$计算的. 本质上, 用于估计泛化误差的个体并未被用来训练分类树集成, 只不过充分利用了训练集所提供的信息.

在实际应用中, 当只有一个数据集可用, 而它必须被同时用来学习分类器和选择最优参数时, 交叉确认法是人们常用的方法. 尽管采用交叉确认法可以得到令人满意的结果, 但它的计算费用往往较高. 假定训练一个基分类树的计算费用为f, 而V折交叉确认法被用来选取K的最优值, 则需要耗费$V\times(p\times B\times f)$才可以得到$K^*$. 而对于上述新提出的方法, 只需$p\times B\times f$就可以得到类似结果, 但其计算费用只是交叉确认法的$1/V$. 当处理的是大规模数据集或者采用的分类树算法比较复杂时, 节省的计算费用将是非常可观的.

4　试验研究

在本节中,采用25个UCI实际数据集 [21]进行试验来验证上节中所提出方法的有效性. 表1列出了试验数据集的主要特征(数据集的样本容量、输入特征个数和类的总数), 这些数据集已经被多次应用于检验和比较集成分类算法的分类效果.　下面的试验是采用Matlab 7.7进行的, 在试验过程中, 分类树算法$\mathscr{C}$采用的

表1 试验数据集的主要特征

Table 1 The main characteristics of the used data sets

数据集	样本容量	特征集大小	类总数	数据集	样本容量	特征集大小	类总数
Abalone	4 177	10	3	Mfeat_mor	2 000	6	10
Australian	690	14	2	Pima	768	8	2
Autompg	398	6	2	Segmentation	2 310	19	7
Balanve	625	4	3	Soybean	136	35	4
Bcs	286	9	2	Sonar	208	60	2
Biomed	194	5	2	Vehicle	846	18	4
Car	1 728	21	4	Votes	435	16	2
Glass	214	10	7	Vowelc	990	12	11
Haberman	306	3	2	Wine	178	13	3
Ionosphere	351	34	2	Wdbc	569	30	2
Iris	150	4	3	Wpbc	194	32	2
Imox	192	8	4	Yeast	1 484	8	10
Liver	345	6	2				

是CART (通过Matlab中的“Treefit”函数实现), 分类树总数$B=500$. 对每个数据集, 由于没有单独的训练集和检验集可供使用, 这里采用3折交叉确认法来估计随机森林算法的泛化误差, 并将试验重复进行10次, 以消除试验过程中对数据进行随机划分所产生的随机效应. 试验的具体过程如下.

1)　将给定数据集随机分成大小基本一致的3个集合$\mathscr{D}=\{\mathscr{D}^1,\mathscr{D}^2,\mathscr{D}^3\}$;

2)　对于K的每个取值($K=1,2,\cdots,p$), 按以下步骤进行.

- 将其中一个集合$\mathscr{D}^1$用作检验集$\mathscr{D}_{\rm test}$, 另外两个集合合并在一起用作训练集$\mathscr{D}_{\rm train}=\{\mathscr{D}^2,\mathscr{D}^3\}$;
- 应用随机森林算法于$\mathscr{D}_{\rm train}$上训练分类树集成C^*, 按照第3节中的方法采用out-of-bag样本估计相应的泛化误差$\varepsilon_{{\rm oob},1}(K)$(在后文中称为out-of-bag误差, oob error), 同时也计算C^*在检验集$\mathscr{D}_{\rm test}$上的误差$\varepsilon_{{\rm test},1}(K)$(检验误差, test error);
- 分别用集合$\mathscr{D}^2$和$\mathscr{D}^3$作为检验集, 而$\{\mathscr{D}^1,\mathscr{D}^3\}$和$\{\mathscr{D}^1,\mathscr{D}^2\}$ 分别用作相应的训练集, 重复上述步骤, 计算出out-of-bag误差$\varepsilon_{{\rm oob},2}(K)$, $\varepsilon_{{\rm oob},3}(K)$和检验误差$\varepsilon_{{\rm test},2}(K)$, $\varepsilon_{{\rm test},3}(K)$.
- 将得到的out-of-bag误差和检验误差进行平均, 得到

$$\varepsilon_{\rm oob}(K)=\frac{1}{3}\sum_{i=1}^{3}\varepsilon_{{\rm oob},i}(K),\ \varepsilon_{\rm test}(K)=\frac{1}{3}\sum_{i=1}^{3}\varepsilon_{{\rm test},i}(K)$$

3) 将上述步骤1)和2)重复进行10次, 对试验结果进行平均.

图1中给出了在Glass, Votes, Wine和Yeast数据集上, 随机森林算法的平均误差(包括out-of-bag误差和检验误差)随其超参数K的变化趋势图。在其他数据集上的试验结果基本类似, 由于本文篇幅的限制, 在此未给出. 在图1的每个子图中, 横坐标表示随机森林算法中超参数K 的不同取值，纵坐标表示利用10次3折交叉确认法估计的随机森林算法在相应的数据集上的分类误差, 包括检验误差(图中的实线)和out-of-bag误差(图中的虚线).

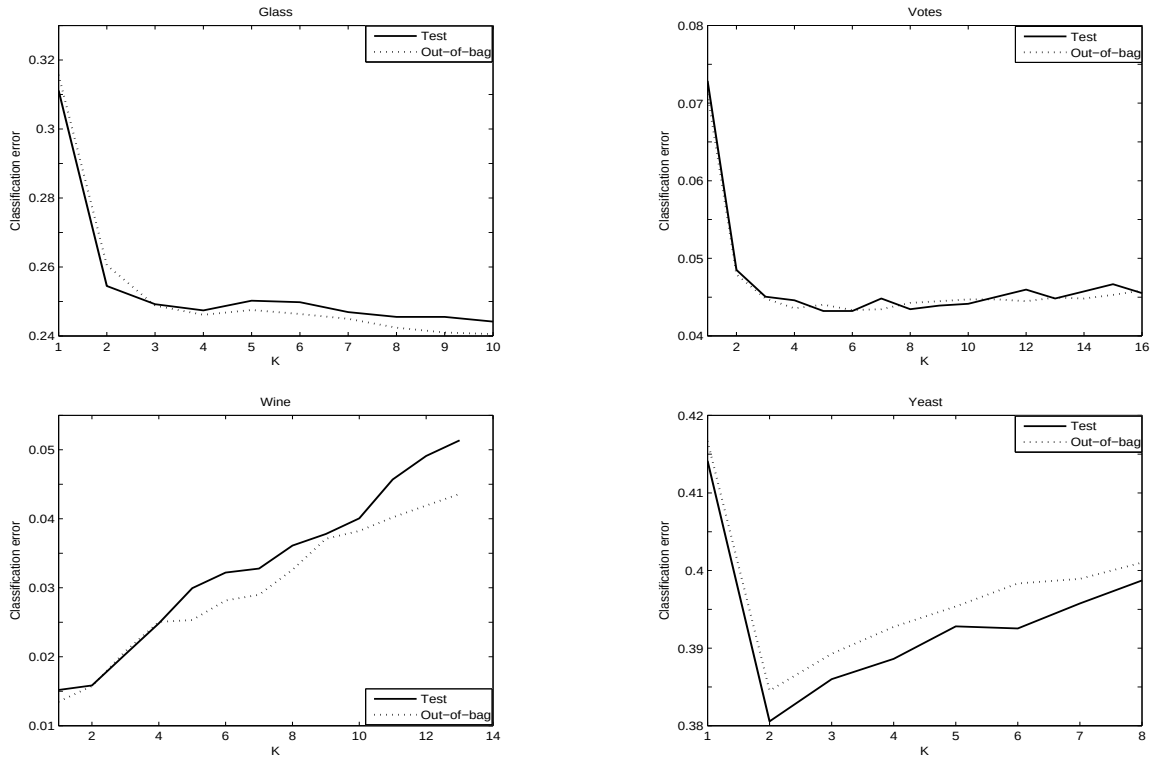

图1 随机森林的平均误差随其超参数K的变化趋势

Fig. 1 The dependence of the classification error (test error and Out-of-bag error) of random forest on the hyper-parameter K

从图1中可以看出, 基于out-of-bag样本估计的误差与检验误差是基本吻合的, 两个误差随着K 取值变化的趋势也完全相同, 这说明采用out-of-bag误差来估计超参数K的最优取值是可行的.

于是, 在每个数据集上,选取使out-of-bag误差达到最小时对应的K^*作为参数K的最优值, 将K^* 对应的检验误差与其他文献[6,13–14]中常用的K值所对应的检验误差进行比较. 由于K的取值只能为整数, 故对K的某些取值进行了取整运算, 其中, $\lfloor A\rfloor$表示小于A的最大整数. 表2列出了在每个数据集上, 随机森林算法采用不同K值所得到的检验误差在10次试验中的均值与标准差.

在表2中, 黑体表示在每个数据集上达到最小检验误差的K值对应的结果. 同时, 还采用单边的成对t检验方法[5]检验了K^*对应的试验结果是否显著地优于K 的其他取值所得的结果. 对于每个数据集, 表中的"•"表示在显著性水平$\alpha=0.05$时, K^*的试验结果显著地优于相应K值的试验结果, 空白则表示K^*与K值的结果之间差别并不显著.表中黑体表示每个数据集上的最优预测结果,小括号中的数字表示超参数K相应算法在试验中的实际取值.

表2 参数K的不同取值所对应的检验误差的均值与标准差

Table 2 The means and standard deviations of test errors corresponding to different values of K

数据集	K^*	1		$\lfloor\log_2(\mathrm{p})+1\rfloor$		$\lfloor\sqrt{p}\rfloor$		p	
Abalone	**34.69±0.37**(2)	37.11±0.31	•	35.12±0.53(4)	•	34.90±0.58(3)		35.94±0.42(10)	•
Autompg	**10.93±0.90**(2)	11.13±0.86		11.18±1.02(3)		**10.93±0.90**(2)		11.00±0.89(6)	
Australian	**13.12±0.59**(2)	13.51±0.40	•	13.18±0.41(4)		13.22±0.56(3)		13.44±0.78(14)	
Balance	**11.74±1.08**(1)	**11.74±1.08**		14.18±0.97(3)	•	13.10±0.83(2)	•	14.90±0.98(4)	•
Bcs	**25.80±1.26**(1)	**25.80±1.26**		27.97±1.61(4)	•	27.62±1.36(3)	•	28.81±1.39(9)	•
Biomed	**11.30±0.90**(1)	**11.30±0.90**		11.51±1.01(3)		12.17±0.85(2)	•	11.25±1.10(5)	
Car	**3.85±0.31**(15)	29.98±0.15	•	4.88±0.36(5)	•	5.84±0.45(4)	•	4.04±0.35(21)	
Glass	**24.42±0.57**(10)	31.12±1.06	•	24.74±0.40(4)		24.92±0.74(3)		**24.42±0.57**(10)	
Haberman	**27.19±0.97**(1)	**27.19±0.97**		28.92±2.59(2)	•	27.19±0.97(1)		29.77±1.58(3)	•
Ionosphere	7.58±0.83(9)	8.12±0.45	•	7.21±0.66(6)		**7.21±0.50**(5)		8.35±0.72(34)	•
Iris	**4.70±1.11**(2)	5.29±0.84		4.83±1.11(3)		**4.70±1.11**(2)		4.82±1.06(4)	
Imox	5.78±0.90(3)	6.15±0.88		5.89±0.98(4)		**5.42±1.05**(2)		6.56±1.28(8)	
Liver	27.05±2.05(1)	27.05±2.05		27.60±1.75(3)		**26.97±2.20**(2)		28.51±1.42(6)	•
Mfeat_mor	**28.65±0.54**(3)	30.38±0.41	•	**28.65±0.54**(3)		28.89±0.38(2)		29.26±0.61(6)	•
Pima	**23.89±1.05**(2)	24.40±0.70		23.97±0.84(4)		**23.89±1.05**(2)		24.41±0.92(8)	
Segmentation	**2.54±0.20**(6)	4.53±0.24	•	2.55±0.20(5)		**2.51±0.17**(4)		3.13±0.13(19)	•
Soybean	**11.00±1.21**(5)	14.30±2.58	•	11.29±1.23(6)		**11.00±1.21**(5)		15.65±2.38(35)	•
Sonar	**18.85±1.58**(3)	19.61±1.33		19.04±1.36(6)		19.19±2.23(7)		22.03±2.27(60)	•
Vehicle	25.50±1.21(6)	26.58±1.08	•	**25.12±1.14**(5)		**25.25±1.14**(4)		26.12±1.38(18)	
Votes	**4.32±0.26**(6)	7.29±0.29	•	4.32±0.30(5)		4.46±0.31(4)		4.55±0.76(16)	
Vowelc	**7.01±0.86**(2)	10.59±1.31	•	7.91±0.82(4)	•	7.30±0.82(3)		13.28±0.74(12)	•
Wine	**1.52±0.45**(1)	**1.52±0.45**		2.48±0.54(4)	•	2.04±0.55(3)	•	5.14±1.94(13)	•
Wdbc	4.59±0.66(10)	4.59±0.56		**4.53±0.55**(5)		**4.53±0.55**(5)		5.04±0.78(30)	
Wpbc	**23.45±0.37**(1)	**23.45±0.37**		24.44±0.86(6)	•	24.28±0.90(5)	•	26.45±1.72(32)	•
Yeast	**38.06±0.75**(2)	41.41±0.53	•	38.86±0.86(4)	•	**38.06±0.75**(3)		39.87±1.05(8)	•
Win/Tie/Loss		12/13/0		9/16/0		6/19/0		14/11/0	

表2的最后一行给出了在所有数据集上“Win/Tie/Loss”统计量的结果, 其中的三个数字分别表示K^* 对应结果显著优于、差别不显著和显著差于其他K值所对应结果的数据集总数. 以K的取值为p 时对应的结果为例(表2中最后一列), “14/11/0”表示在所考查的25个数据集中, 利用本文提出的方法所选取的K^*使得随机森林算法的分类效果在14个数据集上显著优于$K=p$ 时的效果, 两种方法之间的效果在11个数据集上没有显著差别, 而令$K=p$ 时随机森林算法的效果没有在一个数据集上显著优于本文提出的方法. 从表2中的试验结果可以看出, K^*对应的平均检验误差在多数数据集(19个)上都是最小的, 且在任何情况下, 其结果都不显著差于K的其他取值. 在12个数据集上, K^*显著优于每次随机选取1个特征作为分裂变量的结果, 在剩余的13个数据集上, 两种方法的差别并不显著. 对于另外两种常用的取值$\lfloor\log_2(p)+1\rfloor$和$\lfloor\sqrt{p}\rfloor$, 利用out-of-bag方法选取的$K^*$分别在9个和6个数据集上显著优于其相应结果. 当$K=p$时, 随机森林算法就是Bagging算法. 因此, 从表2的最后两列可以看出采用K^* 的随机森林算法在14个数据集上都显著优于Bagging算法的分类效果

5 结束语

随机森林作为一种有效的分类树集成算法, 在使用过程中需要确定其超参数K的最优取值, 以使其预测

效果达到最优. 尽管目前有一些方法可以用于选取K的值, 但大多都是一些启发式的方法或需要很大的计算量. 本文在不额外增加计算复杂性的前提下, 提出了一种基于out-of-bag样本估计其超参数取值的方法. 仿真试验的结果表明, 利用文中提出的方法所选取的随机森林的超参数K在多数情况下都是最优的, 从而为随机森林算法在实际问题中的应用奠定了一定基础.

参考文献:

[1] Oza N C, Tumer K. Classifier ensembles: Select real-world applications[J]. Information Fusion, 2008, 9(1): 4–20.
[2] Hu Q H, Yu D R, Xie Z X, et al. EROS: Ensemble rough subspaces[J]. Pattern Recognition, 2007, 40(12): 3728–3739.
[3] Wang X Z, Zhai J H, Lu S X. Induction of multiple fuzzy decision trees based on rough set technique[J]. Information Sciences, 2008, 178(16): 3188–3202.
[4] Kucheva L I. Combining Pattern Classifiers, Methods and Algorithms[M]. Hoboken: Wiley Interscience, 2004.
[5] 张春霞. 集成学习中有关算法的研究[D]. 西安: 西安交通大学, 2009.
Zhang Chunxia. Research on some algorithms in ensemble learning[D]. Xi'an: Xi'an Jiaotong University, 2009. (in Chinese)
[6] Breiman L. Random forest[J]. Machine Learning, 2001, 45(1): 5–32.
[7] 贾富仓, 李 华. 基于随机森林的多谱磁共振图像分割[J]. 计算机工程, 2005, 31(10): 159–161.
Jia Fucang, Li Hua. Multi-spectral magnetic resonance image segmentation using random forests [J]. Computer Engineering, 2005, 31(10): 159-161. (in Chinese)
[8] 袁 敏, 胡秀珍. 随机森林方法预测膜蛋白类型[J]. 生物物理学报, 2009, 25(5): 349–355.
Yuan Min, Hu Xiuzhen. Predicting membrane protein types using the random forests algorithm[J]. Acta Biophysica Sinica, 2009, 25(5): 349–355. (in Chinese)
[9] 庄进发, 罗 键, 彭彦卿, 等. 基于改进随机森林的故障诊断方法研究[J]. 计算机集成制造系统, 2009, 15(4): 777–785.
Zhuang Jinfa, Luo Jian, Peng Yanqing, et al. Fault diagnosis method based on modified random forests[J]. Computer Integrated Manusfacturing Systems, 2009, 15(4): 777–785. (in Chinese)
[10] Genuer R, Poggi J M, Tuleau-Malot C. Variable selection using random forests[J]. Pattern Recognition Letters, 2010, 31(14): 2225–2236.
[11] Efron B, Tibshirani R. An Introduction to the Bootstrap[M]. New York: Chapman & Hall, 1993.
[12] Breiman L, Friedman J, Olshen R, et al. Classification and Regression Trees[M]. New York: Chapman & Hall, 1984.
[13] Paňov P, Džeroski S. Combining bagging and random subspaces to create better ensembles[C]//Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, 2007, 4723: 118–129.
[14] Bernard S, Heutte L, Adam S. Influence of Hyperparameters on Random Forest Accuracy[C]// Proceedings of the 8th International Workshop on Multiple Classifier Systems, Berlin, Heidelberg: Springer-Verlag, 2009: 171–180.
[15] Breiman L. Out-of-bag Estimation[R]. Berkeley: Statistics Department, University of California, 1996.
[16] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123–140.
[17] 李 毓, 徐成贤. 修剪Bagging集成的方法及其应用[J]. 系统工程理论与实践, 2008, 7: 105–110.
Li Yu, Xu Chengxian. A method for pruning bagging ensembles and its applications[J]. Systems Engineering : Theory & Practice, 2008, 7: 105–110. (in Chinese)
[18] Bylander T. Estimating generalization error on two-class datasets using out-of-bag estimates[J]. Machine Learning, 2002, 48(1-3): 287–297.
[19] Hothorn T, Lausen B. Double-bagging: Combining classifiers by bootstrap aggregation[J]. Pattern Recognition, 2003, 36(6): 1303–1309.
[20] Martínez-Muñoz G, Suárez A. Out-of-bag estimation of the optimal sample size in bagging[J]. Pattern Recognition, 2010, 43(1): 143–152.
[21] Asuncion A, Newman D J. UCI machine learning repository[DB/OL]. Irvine: University of California, School of Information and Computer Science, http://www.ics.uci.edu/˜mlearn/MLRepository.html, 2007.

作者简介:

李 毓 (1963—), 男, 河南信阳人, 博士,教授, 研究方向: 金融工程、优化与模式识别, E-mail: ly632003@yahoo.com.cn;
张春霞 (1980—), 女, 河南荥阳人, 博士, 讲师, 研究方向: 模式分类、集成学习.