基于贡献增益的森林剪枝

郭华平 范 明
(郑州大学信息工程学院 郑州 450052)

摘 要 基于决策树的组合分类器可以看作一个森林。提出了一种森林剪枝算法来对森林进行剪枝,以简化组合分类器的结构,并提高其分类准确率。传统的决策树剪枝只考虑剪枝对单棵决策树的影响,而森林剪枝则把所有决策树看作一个整体,更加关注剪枝对组合分类器的性能影响。为了确定森林的哪些分枝可以被剪枝,提出一种称作贡献增益的度量。子树的贡献增益不仅与它所在的决策树的分类准确率有关,而且也与诸决策树的差异性有关,因此它较好地度量了一个结点扩展为一棵子树对组合分类器分类准确率的提高程度。借助于贡献增益,设计了一种基于结点贡献增益的森林剪枝算法FTCG。实验表明,无论森林是基于某种算法(如bagging)构建的还是某种组合分类器选择算法(如EPIC[1])的结果,无论每棵决策树是未剪枝的还是剪枝后的,FTCG都能进一步降低每棵决策树的规模,并且在大部分数据集上显著提高了剪枝后的组合分类器的分类准确率。

关键词 森林剪枝,组合选择,贡献增益
中图法分类号 TP181 **文献标识码** A

Forest Thinning via Contribution Gain

GUO Hua-ping FAN Ming
(School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China)

Abstract An ensemble consisting of decision trees can be treated as a forest. We proposed a new strategy called forest thinning to reduce the ensemble size and improve its accuracy. Unlike traditional decision tree pruning methods, forest thinning treats all decision trees in a forest as a whole and try to evaluate the performance influence on the ensemble when a certain branch is pruned. To determine which branches should be pruned, we proposed a new metric called contribution gain. The contribution gain of a subtree is related not only to the accuracy of the host tree, but also to the diversity of trees in the ensemble, so it reasonably well measures how much improvement of the ensemble accuracy can be achieved when a subtree is pruned. With the contribution gain, we designed a forest thinning algorithm named FTCG (Forest Thinning via Contribution Gain). Our experiments show that forest thinning can significantly reduce forests structure complexity and improve their accuracy in most of data sets, no matter ensembles are constructed by a certain algorithm such as bagging, or obtained by an ensemble selection algorithm such as EPIC [1], no matter whether each decision tree is pruned or unpruned.

Keywords Forest thinning, Ensemble selection, Contribution gain

1 引言

在过去的十几年里,组合分类方法一直是机器学习和数据挖掘中非常活跃的研究领域。已经提出了许多建立组合分类器的方法,例如bagging[2]、boosting[3]以及旋转森林[4]等。大量的理论与实践结果表明,给定相同训练信息,组合分类器往往表现出比单个分类器更好的泛化能力。

然而,大部分组合分类方法都存在一个共同的问题:倾向于构建大量的基分类器。大量基分类器不但需要大量的存储空间还大幅度降低了分类器的预测速度。近几年,为了降低组合分类器的结构复杂性和提高其性能,已经开展了许多工作,这些工作主要集中在基分类器的选择上,即:从组合分类器成员中选择一个最优或次优的子集用于预测未知样本类标号[1,5-7]。相关的研究表明,一个精心选择的子组合分类器不仅可以降低组合分类器规模,而且可以提高它的分类准确率。

事实上,当基分类器是决策树时,还可以通过决策树剪枝来简化组合分类器以及提高它的分类准确率。对组合分类器中的诸决策树进行剪枝有两种方法:(1)不考虑对整体的影响,分别对每棵树进行剪枝;(2)从整体考虑出发,反复对每棵树进行剪枝,以期达到最好的整体效果。

对于第1种方法,单棵决策树的剪枝已有广泛的研究,如EBP[8]等。剪枝能够简化模型是毋庸置疑的,但是,能否提高结果模型的分类准确率一直存在争议[9]。按照一种广泛接受的说法,影响组合分类器分类准确率的主要因素是基分离器

到稿日期:2013-01-29 返修日期:2013-06-26 本文受国家自然科学基金项目(偏好学习的若干关键技术研究,60901078)资助。
郭华平(1982—),男,博士生,主要研究方向为数据挖掘、机器学习,E-mail:hpguo.gm@gmail.com;范 明(1948—),男,教授,博士生导师,CCF高级会员,主要研究方向为数据挖掘、机器学习、数据库。

之间的差异性和基分类器分类准确率。单棵决策树的剪枝完全不考虑其它决策树，根本无法保证有助于提高基分类器的差异性。由于剪枝对基分类器分类准确率的提高也存在疑问，试图通过单棵决策树的剪枝来提高组合分类器的分类准确率，结果必然是好坏参半[10]。

第2种方法无疑是一种好方法，因为对组合分类器而言，我们期望的正是整体效果好，而不是单棵决策树的分类准确率。遗憾的是，这方面几乎看不到已发表的研究成果。主要原因可能是难以确定对森林中的哪些树的哪些分枝进行剪枝。本文将讨论第2种方法，我们称之为森林剪枝。

森林剪枝的关键问题是如何建立一种度量标准，以度量剪掉某棵决策树的某个子树对整个组合分类器的分类准确率的影响，从而决定哪些子树可以用树叶结点取代。传统的决策树剪枝的度量只考虑剪枝对单棵决策树的影响，并不考虑对组合分类器的影响，不能用于森林剪枝。因此我们需要为森林剪枝建立新的度量标准。本文的主要贡献是：

- 提出了一种称作结点贡献增益(Contribution Gain, ConGain)的度量，用来评估决策树 T 的非终端结点 v 生长成一棵子树对整个组合分类器分类准确率的提高程度。
- 基于贡献增益，提出了一种森林剪枝算法 FTCG，对基于决策树的组合分类器进行剪枝。其中，组合分类器可以是基于某种算法(如 bagging)构建的，也可以是某种组合分类器选择算法(如 EPIC)的结果；每棵决策树可以是未剪枝的，也可以是剪枝后的。

我们的实验表明，FTCG 显著地降低了每棵决策树的规模，并且在大多数数据集上都显著提高了结果模型的泛化性能。此外，我们的实验结果还表明本文提出的结点贡献增益较好地度量了结点在决策树中的生长对整个组合分类器分类准确率的影响。

本文第2节给出问题的形式描述，并用一个例子说明森林剪枝的动机；第3节给出结点贡献增益的定义、FTCG 算法的基本思想和伪代码；第4节报告和分析我们的两组实验结果；最后，用简单的评述和进一步工作结束本文。

2 问题描述与动机

2.1 问题描述

设 $D=\{(\boldsymbol{x}_i, y_i)\}$ 是训练数据集，其中 x_i 给出第 i 个实例的诸属性值，$y_i \in \{1,\cdots,K\}$ 为类标记。假定我们已经在训练数据集 D 上建立了一个组合分类器 $F=\{T_1,\cdots,T_m\}$，其中每个 T_i 都是一棵决策树。F 也可以看作一个森林。本文中把组合分类器和森林视为同义词，因为我们只讨论基于决策树的组合分类器。

设 $T \in F$ 为任意决策树，v 为 T 的任意结点。令 $E(v) \in D$ 为从 $root(T)$ 沿着 T 的路径到达结点 v 的训练实例的集合。假定决策树 T 的每个结点 v 都包含一个向量 $(p_1^v,\cdots,p_K^v)$，其中 p_k^v 是 $E(v)$ 中样本属于类 k 的概率，它是 $E(v)$ 中属于类 k 的实例所占的比例。如果 v 是 T_i 的树叶结点，并且实例 $\boldsymbol{x}_j \in E(v)$，则把 p_k^v 记作 $p_{jk}^{(i)}$，并称对于实例 $\boldsymbol{x}_j$，T_i 返回向量 $(p_{j1}^{(i)},\cdots,p_{jK}^{(i)})$，指明 $\boldsymbol{x}_j$ 属于类 k 的概率为 $p_{jk}^{(i)}$。T_i 将预测 $\boldsymbol{x}_j$ 属于类 $T_i(\boldsymbol{x}_j)$，其中 $T_i(\boldsymbol{x}_j)=\arg\max_k(p_{jk}^{(i)})$。

上述假定是合理的，因为所有的决策树分类算法即使未提供这些概率，稍加调整也都可以提供这些信息。

类似地，对于每个待分类实例 $\boldsymbol{x}_j$，组合分类器 F 也返回一个向量 $(p_{j1},\cdots,p_{jK})$，指明 $\boldsymbol{x}_j$ 属于类 k 的概率为 p_{jk}，其中

$$p_{jk}=\frac{1}{M}\sum_{i=1}^{m}p_{jk}^{(i)},\ k=1,\cdots,K \tag{1}$$

组合分类器 F 将预测 $\boldsymbol{x}_j$ 属于类 $F(\boldsymbol{x}_j)$，其中 $F(x_j)=\arg\max_k(p_{jk})$。

我们的问题是：给定一个森林 $F=\{T_1,\cdots,T_M\}$，如何对诸 T_i 进行剪枝，以降低组合分类器 F 的结构复杂度，并同时得到类似或更高的分类准确率。这里，F 可以是基于某种算法(如 bagging)构建的，也可以是某种组合分类器选择算法(如 EPIC)的结果。

2.2 动机

首先，我们看一个例子，它展示了森林剪枝的可能性。

例1 设 $F=\{T_0,T_1,\cdots,T_9\}$ 是包含10棵决策树的组合分类器。考虑图1所示的决策树 T_0，其中，$p_1^v=0.60$，$p_2^v=0.40$；$p_1^{v_1}=1.00$，$p_2^{v_1}=0.00$；$p_1^{v_2}=0.20$，$p_2^{v_2}=0.80$。

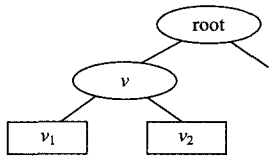


图1 决策树 T_0，其中，v 是测试结点，v_1 和 v_2 是两个叶子结点

设10个检验实例 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_9$ 到达决策树 T_0 的结点 v，其中 $\boldsymbol{x}_0,\cdots,\boldsymbol{x}_5$ 属于类1，$\boldsymbol{x}_6,\cdots,\boldsymbol{x}_9$ 属于类2。又设 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_4$ 到达叶节点 v_1，$\boldsymbol{x}_5,\cdots,\boldsymbol{x}_9$ 到达叶节点 v_2。

显然，就 T_0 而言，不能剪掉 v 的子女，因为这将导致更多的实例误分类。

设组合分类器 F 对实例 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_9$ 返回如下概率：

$p_{01}=0.65$，$p_{11}=0.70$，$p_{21}=0.70$，$p_{31}=0.65$，$p_{41}=0.80$

$p_{51}=0.49$，$p_{61}=0.30$，$p_{71}=0.19$，$p_{81}=0.20$，$p_{91}=0.30$

$p_{02}=0.35$，$p_{12}=0.30$，$p_{22}=0.30$，$p_{32}=0.35$，$p_{42}=0.20$

$p_{52}=0.51$，$p_{62}=0.70$，$p_{72}=0.81$，$p_{82}=0.80$，$p_{92}=0.70$

其中 p_{ij} 表示组合分类器 F 预测实例 i 应分类为类 j 的概率。由上面的组合分类器的预测结果可知，F 把 $\boldsymbol{x}_6$ 错误分类，而对其它9个实例都正确分类。

令 T_0' 是 T_0 剪去 v 的子女得到的决策树，而 $F'=\{T_0',T_1,\cdots,T_9\}$。只需简单的计算就可以得到组合分类器 F' 对实例 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_9$ 返回的如下概率：

$p_{01}=0.60$，$p_{11}=0.65$，$p_{21}=0.65$，$p_{31}=0.65$，$p_{41}=0.75$

$p_{51}=0.52$，$p_{61}=0.33$，$p_{71}=0.22$，$p_{81}=0.23$，$p_{91}=0.33$

$p_{02}=0.40$，$p_{12}=0.35$，$p_{22}=0.35$，$p_{32}=0.35$，$p_{42}=0.25$

$p_{52}=0.48$，$p_{62}=0.67$，$p_{72}=0.78$，$p_{82}=0.77$，$p_{92}=0.67$

容易看出，F' 对所有的实例都正确分类。

这个例子表明：单独考虑，每棵决策树可能都不适合再进行剪枝。但是从整体考虑，仍然有可能对每棵决策树进行剪枝，并且这种剪枝不但不会降低而且可能会提高组合分类器的分类准确率。

尽管这个例子是我们构建的，但是如果深入考察组合算法建立的实际组合分类器，类似的情况随处可见。正是这种

观察促使我们考虑森林剪枝。然而，把可能性变成可行性仍然有许多工作要做。下面将对此进行深入讨论。

3 森林剪枝算法 FTCG

FTCG 算法是基于一种称作结点贡献度增益(Contribution Gain，简记为 ConGain)的度量的算法。本节，首先引入 ConGain 的定义，简述 FTCG 算法的基本思想；然后，进一步讨论一些相关概念的细节，最后，给出 FTCG 算法的伪代码，并分析该算法复杂度。

3.1 贡献度增益与算法思想

为了避免过早陷入细节，假定我们已经定义了 $Con(v,F,\boldsymbol{x}_j)$，它是组合分类器 F 对 $\boldsymbol{x}_j$ 分类时，决策树 T 的结点 v 的贡献。如果 $\boldsymbol{x}_j\in E(v)$，则 $Con(v,F,\boldsymbol{x}_j)=0$。如果 $\boldsymbol{x}_j\in E(v)$，则 $Con(v,F,\boldsymbol{x}_j)$ 的定义在 3.2 节详细讨论。

设 $T\in F$ 是任意决策树，T 的结点 v 对组合分类器 F 的贡献(简称结点 v 的贡献，记作 $Con(v,F)$)，定义为

$$Con(v,F)=\sum_{\boldsymbol{x}_j\in D}Con(v,F,\boldsymbol{x}_j)=\sum_{\boldsymbol{x}_j\in E(v)}Con(v,F,\boldsymbol{x}_j) \quad (2)$$

式中，$E(v)$是从 $root(T)$沿着 T 的路径到达结点 v 的训练集 D 中实例的集合。$Con(v,F)$体现了 T 的结点 v 对 F 的分类准确率的影响。

设 $L(v)$为 T 的子树 $subtree(v)$中的叶结点的集合。T 的子树 $subtree(v)$对组合分类器 F 的贡献(简称子树 $subtree(v)$的贡献)，记作 $Con(subtree(v),F)$，定义为

$$Con(subtree(v),F)=\sum_{v'\in L(v)}Con(v',F) \quad (3)$$

它是 $subtree(v)$中叶结点的贡献和。

设 v 为 T 的非树叶结点，v 对组合分类器 F 的贡献增益(简称结点 v 的贡献增益)，记作 $ConGain(v,F)$，定义为子树 $subtree(v)$对组合分类器 F 的贡献与结点 v 对组合分类器 F 的贡献之差，即

$$ConGain(v,F)=Cons(subtree(v),F)-Con(v,F) \quad (4)$$

结点 v 的贡献增益也可以看作子树 $subtree(v)$的贡献增益，它体现了 v 扩展为子树 $subtree(v)$对组合分类器 F 的分类准确率的提高的影响。如果 $ConGain(v,F)>0$，则这一扩展有助于提高 F 的分类准确率，否则无助于提高、甚至降低 F 的分类准确率。

FTCG 算法的思想非常简单：对于每棵树 T，在训练集上计算 T 的每个非树叶结点 v 的贡献增益 $ConGain(v,F)$；如果 $ConGain(v,F)$小于某个阈值，则剪掉 $subtree(v)$，把 v 转换成树叶。然后，继续这一过程，直到任何决策树都不能再剪枝。

3.2 计算 $Con(v,F,\boldsymbol{x}_j)$

在考虑基分类器选择时，Partalas 等根据基分类器和组合分类器对实例$(\boldsymbol{x}_i,y_i)$的分类正确与否，把实例分成如下 4 种情况[6]：(1)$e_{tf}:h(\boldsymbol{x}_i)=y_i\wedge S(\boldsymbol{x}_i)\neq y_i$，(2)$e_{tt}:h(\boldsymbol{x}_i)=y_i\wedge S(\boldsymbol{x}_i)=y_i$，(3)$e_{ft}:h(\boldsymbol{x}_i)\neq y_i\wedge S(\boldsymbol{x}_i)=y_i$，(4)$e_{ft}:h(\boldsymbol{x}_i)\neq y_i\wedge S(\boldsymbol{x}_i)\neq y_i$。其中，$h$ 表示基分类器，对应于本文的决策树 T_i；S 表示组合分类器，对应于本文的 F。他们相信这种分类对设计基分类器差异性度量至关重要。

基于这种分类，在组合分类器对第 j 个实例分类时，Lu 等提出使用个体贡献 $IC_i^{(j)}$ 度量第 i 个基分类器的贡献[1]；Partalas 等提出一种叫做不确定性加权准确率 $UWA_D(h,S,i)$度量基分类器 h 的贡献[6]。这两种度量都可以稍加修改，用于定义 $Con(v,F,\boldsymbol{x}_j)$。但是，我们的实验表明，其结果并不能令人满意。

类似于 Partalas 等的做法，我们定义

$$\begin{aligned}
e_{tf}(v)&=\{\boldsymbol{x}_j\mid\boldsymbol{x}_j\in E(v)\wedge T_i(\boldsymbol{x}_j)=y_j\wedge F(\boldsymbol{x}_j)\neq y_j\}\\
e_{tt}(v)&=\{\boldsymbol{x}_j\mid\boldsymbol{x}_j\in E(v)\wedge T_i(\boldsymbol{x}_j)=y_j\wedge F(\boldsymbol{x}_j)=y_j\}\\
e_{ft}(v)&=\{\boldsymbol{x}_j\mid\boldsymbol{x}_j\in E(v)\wedge T_i(\boldsymbol{x}_j)\neq y_j\wedge F(\boldsymbol{x}_j)=y_j\}\\
e_{ff}(v)&=\{\boldsymbol{x}_j\mid\boldsymbol{x}_j\in E(v)\wedge T_i(\boldsymbol{x}_j)\neq y_j\wedge F(\boldsymbol{x}_j)\neq y_j\}
\end{aligned} \quad (5)$$

在下面的讨论中，假定 v 是决策树 T 的结点，$\boldsymbol{x}_j\in E(v)$。令 $f_m=\mathrm{argmax}_k(p_{j1},\cdots,p_{jK})$，$f_s=\mathrm{argmax}_k(\{p_{j1},\cdots,p_{jK}\}-p_{jf_m})$，即 f_m 和 f_s 分别是$\{p_{j1},\cdots,p_{jK}\}$中最大元素和次大元素下标。类似地，令 $t_m=\mathrm{argmax}_k(p_1^v,\cdots,p_K^v)$，$t_s=\mathrm{argmax}_k(\{p_1^v,\cdots,p_K^v\}-p_{t_m}^v)$，即 t_m 和 t_s 分别是$\{p_1^v,\cdots,p_K^v\}$中最大元素和次大元素的下标。显然，f_m 是组合分类器 F 对 x_j 的类预测。当 v 是 T 的树叶结点时，t_m 就是决策树 T 对 x_j 的类预测；当 v 是 T 的非终端结点时，t_m 是 T'对 x_j 的类预测，其中 T'是从 T 中剪掉 $subtree(v)$得到的决策树。为方便起见，称 t_m 是结点 v 的类预测。

我们将对式(5)的 4 种情况，分别定义 $Con(v,F,\boldsymbol{x}_j)$。当 $\boldsymbol{x}_j\in e_{tf}(v)$或 $\boldsymbol{x}_j\in e_{tt}(v)$时，$\boldsymbol{x}_j$ 被结点 v 正确预测，因此应当有 $Con(v,F,\boldsymbol{x}_j)\geqslant 0$。而当 $\boldsymbol{x}_j\in e_{ft}(v)$或 $\boldsymbol{x}_j\in e_{ff}(v)$时，$\boldsymbol{x}_j$ 被结点 v 错误预测，因此应当有 $Con(v,F,\boldsymbol{x}_j)<0$。

1. 对于 $\boldsymbol{x}_j\in e_{tf}(v)$，我们定义

$$Con(v,F,\boldsymbol{x}_j)=\frac{p_{t_m}^v-p_{f_m}^v}{M(p_{jf_m}-p_{jt_m}+\frac{1}{M})} \quad (6)$$

式中，M 表示 F 中基分类器个数。由于 $p_{t_m}^v\geqslant p_{f_m}^v$，$p_{jf_m}\geqslant p_{jt_m}$，容易证明上式满足 $0\leqslant Con(v,F,\boldsymbol{x}_j)\leqslant 1$。注意，当 $\boldsymbol{x}_j\in e_{tf}(v)$时，$t_m=y_j$ 但 $f_m\neq y_j$。由式(1)可知，$p_{t_m}^v/M$ 是结点 v 对 p_{jt_m}(F 正确地预测 $\boldsymbol{x}_j$ 属于 $t_m=y_j$ 的概率)的贡献，而 $p_{f_m}^v/M$ 是结点 v 对 p_{jf_m}(F 错误地预测 $\boldsymbol{x}_j$ 属于 $f_m\neq y_j$ 的概率)的贡献，因此 $(p_{t_m}^v-p_{f_m}^v)/M$ 可以看作 F 预测 x_j 时，结点 v 的净贡献。$1/(p_{jf_m}-p_{jt_m}+\frac{1}{M})$是结点 v 的净贡献权重，表示结点 v 对正确分类 x_j 的重要程度。其中，常数 $1/M$ 可防止分母为 0 或太小。

2. 对于 $\boldsymbol{x}_j\in e_{tt}(v)$，我们定义

$$Con(v,F,\boldsymbol{x}_j)=\frac{p_{t_m}^v-p_{t_s}^v}{M(p_{jf_m}-p_{jf_s}+\frac{1}{M})} \quad (7)$$

式中，$0\leqslant Con(v,F,\boldsymbol{x}_j)\leqslant 1$。在这种情况下，$v$ 和 F 对 $\boldsymbol{x}_j$ 的类预测都是正确的，即 $t_m=f_m=y_j$，于是 $p_{t_m}^v=p_{f_m}^v$。我们用 $(p_{t_m}^v-p_{t_s}^v)/M$ 表示 F 预测 x_j 时结点 v 的净贡献，而用 $1/(p_{jf_m}-p_{jf_s}+\frac{1}{M})$表示 v 的净贡献权重。

3. 对于 $\boldsymbol{x}_j\in e_{ft}(v)$，我们定义

$$Con(v,F,\boldsymbol{x}_j)=-\frac{p_{t_m}^v-p_{f_m}^v}{M(p_{jf_m}-p_{jf_s}+\frac{1}{M})} \quad (8)$$

式中，$-1\leqslant Con(v,F,\boldsymbol{x}_j)\leqslant 0$。这种情况与情况 1 相反，我们用 $-(p_{t_m}^v-p_{f_m}^v)/M$ 表示结点 v 对 F 预测 $\boldsymbol{x}_j$ 的净贡献，而用 $p_{jf_m}-p_{jf_s}$ 表示 v 对 F 正确分类 $\boldsymbol{x}_j$ 的影响程度。

4. 对于 $\boldsymbol{x}_j\in e_{ff}(v)$，我们定义

$$Con(v,F,\boldsymbol{x}_j)=-\frac{p^v_{t_m}-p^v_{y_j}}{M(p_{jf}-p_{jy_j}+\frac{1}{M})} \quad (9)$$

式中，$y_j\in\{1,\cdots,K\}$是 $\boldsymbol{x}_j$ 的实际类标记，$-1\leqslant Con(v,F,\boldsymbol{x}_j)\leqslant 0$。在这种情况下，$v$ 和 F 对 $\boldsymbol{x}_j$ 的类预测都是错误的，即 $t_m\neq y_j$，$f_m\neq y_j$。我们用 $-(p^v_{t_m}-p^v_{y_j})/M$ 表示结点 v 对 F 预测 $\boldsymbol{x}_j$ 的净贡献，而用 $p_{jf_m}-p_{jy_j}$ 表示 v 对 F 正确分类 $\boldsymbol{x}_j$ 的影响程度。

由于 $Con(v,F,\boldsymbol{x}_j)$同时考虑到了(1)$v$ 所在决策树的准确率，(2)F 成员的差异性，因此 $ConGain(v,F,\boldsymbol{x}_j)$也同时考虑到了这两个因素。

3.3 FTCG算法描述

算法 1 给出了 FTCG 算法的伪代码，其中，D 包含 n 个实例的训练数据集，p_{jk} 组合分类器预测 $\boldsymbol{x}_j$ 属于类 k 的概率，p_{ijk} 保存当前正处理的决策树 T_i 对诸实例的预测概率，q_{jk} 保存旧的 p_{ijk} 以调整剪枝决策树 T_i 后森林的预测，C_v 保存结点 v 的贡献，C_c 保存 subtree(v)的贡献。

算法首先计算 F 对每个实例的预测概率(1－2 行)。然后，它处理每棵子树 T_i(3－14 行)。4－10 行把 T_i 对每个实例的预测概率保存到数组 q 中(7 行)，以便 T_i 剪枝后调整 F 对每个实例的类预测，并通过累加计算 T_i 的每个结点 v 的贡献 C_v。其中，第 10 行中的 $Con(v,F,\boldsymbol{x}_j)$依据式(5)的 4 种情况，用式(6)－式(9)之一计算。对 T_i 的剪枝通过调用递归过程 *PruningTree*(v)完成(11 行)。T_i 剪枝后，需要调整 F 对每个实例的预测概率(12－14 行)，因为此时组合分类器 F 已经因 T_i 剪枝而改变。3－14 行的处理可以进行多轮，直至所有的决策树都不能再被剪枝。实践中，该迭代过程只需要进行 2 轮。

算法 1 FTCG

输入：训练集 D，包含 m 个子树的森林 F={T_1,T_2,…,T_m}

输出：剪枝后的森林 F.

方法：

1. for 每个实例 $\mathbf{x}_j\in$D do
2. 评估 p_{jk}，$1\leqslant k\leqslant K$；
3. for 每棵树 $T_i\in$F do
4. for 每个结点 $v\in T_i$ do
5. $C_v\leftarrow 0$；
6. for each $\mathbf{x}_j\in$D do
7. $q_{jk}\leftarrow p_{ijk}$，$1\leqslant k\leqslant K$；
8. 设 P 为 $\mathbf{x}_j$ 经过的路径；
9. for 每个结点 $v\in$P do
10. $C_v\leftarrow C_v+$Con(v,F,$\mathbf{x}_j$)；
11. PruningTree(root(T_i))；//调用子过程
12. for each $\mathbf{x}_j\in$D′
13. $r_{jk}\leftarrow p_{ijk}$，$1\leqslant k\leqslant K$；
14. $p_{jk}\leftarrow p_{jk}-q_{jk}/M+r_{jk}/M$

Procedure PruningTree(v)

1. if v 不是叶子 then
2. $CG\leftarrow C_v$；
3. $C_v\leftarrow 0$；
4. for v 的每个孩子 c do
5. PruningTree(c)；
6. $CG\leftarrow CG-C_c$；
7. $C_v\leftarrow C_v+C_c$；
8. if $CG<\delta$ then
9. 剪枝 subtree(v)并设置 v 为叶子结点。

递归过程 *PruningTree*(v)实现对 *subtree*(v)的剪枝。其与许多传统决策树剪枝算法相似，对一棵决策树的剪枝自底向上进行。在对 *subtree*(v)进行剪枝处理之后，C_v 中存放 *subtree*(v)中所有叶结点的贡献和。这样 $Con(subtree(v),F)$总是等于其子女结点的贡献和。用 T_i 的根结点调用过程 *PruningTree* 本质上是遍历 T_i。遇到叶结点时，它什么都不做。对于非树叶结点 v，需要计算 v 的贡献增益 CG，并在 C_v 中得到 *subtree*(v)中所有叶结点的贡献和(2－7 行)；然后根据 v 的贡献增益 CG 是否小于某个阈值 δ，决定是否剪掉 *subtree*(v)(8－9 行)。

设剪枝集 D 包含 n 个实例，森林 F 包含 m 棵树，$d_{\max}$ 为森林中最深决策树的深度，$|T_i|$为决策树 T_i 的结点数，而 $t_{\max}=\max_{1\leqslant i\leqslant m}(T_i)$。算法 FTCG 的 1－2 行初始化，需要让每棵决策树对每个实例进行分类，其时间不超过 $O(mnd_{\max})$。4－14 行的循环体对每棵树 T_i 执行一次。其中，4－5 行遍历树 T_i，其时间不超过 $O(t_{\max})$；6－10 行的循环需要对每个实例搜索 T_i 的一条路径，其时间不超过 $O(nd_{\max})$；第 11 行对树 T_i 剪枝，主要操作是遍历 T_i，其时间不超过 $O(t_{\max})$；12－14 行扫描长度为 n 的表，其时间为 $O(n)$。由于 $t_{\max}<n$，因此 3－14行的循环的时间不超过 $O(mnd_{\max})$。于是，整个算法的时间不超过 $O(mnd_{\max})$。

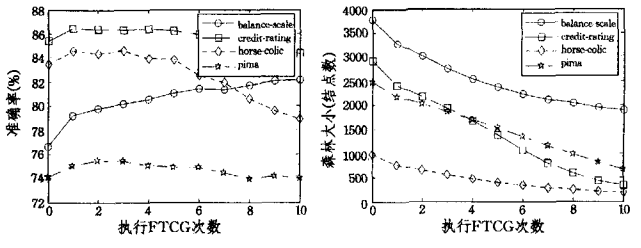
4 实验

4.1 实验设置

论文从 UCI 库[13]中随机选择 18 个数据集测试算法性能。在每个数据集，执行 10 折交叉验证。我们设计了 4 个实验，其中前两个实验分别考察 FTCG 的迭代执行次数和决策树的个数对 FTCG 性能的影响，而后两个实验评估 FTCG 的性能。所有的实验都在 Weka[14]上进行，基分类器使用 J48(C4.5[8]在 Weka 中的实现版本)建立，组合分类器都用 bagging[2]建立，组合分类器选用 EPIC 算法[1]。在算法 1 中，设置 $\delta=0$。

4.2 实验结果

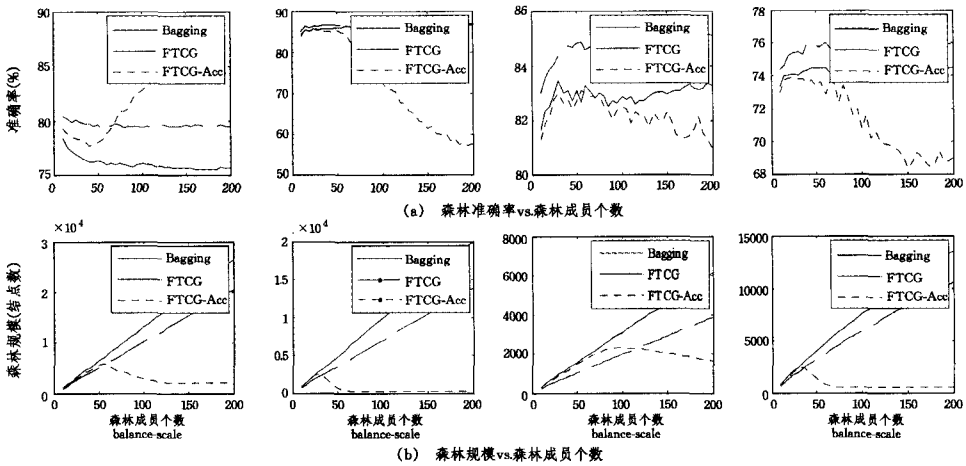
实验 1 旨在考察 FTCG 的多次迭代的剪枝效果，以便确定在后面的实验中如何设置迭代次数 t 的值。实验选取 18 个数据集中的其中 4 个，即 balance-scale，credit-rating，horse-colic 和 pima。先用 bagging 建立包含 30 棵决策树的组合分类器；然后，迭代地执行 FTCG 算法的 3－14 行多次。图 2 显示了 0－10 次迭代执行的结果。其中，图(a)显示随着迭代次数增加，森林准确率变化的趋势，而图(b)显示组合分类器分类规模的变化(使用结点数衡量)。从图 2 可以看出，FTCG 显著地降低了森林的规模，并且显著地提高了组合分类器的准确率。然而，两次迭代之后，FTCG 的性能基本稳定。因此，在后面的实验中，我们设置迭代次数 $t=2$。



(a) 森林准确率 vs. 执行 FTCG 次数　(b) 森林规模 vs. 执行 FTCG 次数

图 2 在 4 个数据集上的结果

实验 2　考察 FTCG 在不同规模的森林上的性能。实验所用数据集与实验 1 相同，组合分类器用 bagging 建立，但森林中的决策树个数从 10 逐渐增加到 200。为了进一步评估森林剪枝的性能，我们测试了 FT-Acc 的性能：FT-Acc 用分类准确率确定是否剪枝一个分枝。实验结果如图 3 所示，其中，上方的 4 个曲线图显示随决策树个数增加，剪枝前后组合分类器的准确率的比较，而下方的 4 个曲线图显示了剪枝前后森林结点个数的比较。正如图 3 所示，对于每个数据集，FTCG 性能显著优于 bagging 且 FTCG 剪枝 bagging 的比例基本稳定。另外，FT-Acc 大幅剪枝 bagging，而其准确率也随着规模的增加而大幅度下降(除 balance-scale 外)。这表明直接使用准确率作为衡量标准剪枝森林容易产生过拟合问题。一个简单的处理该问题的方法是使用交叉验证来确定是否剪枝决策树分枝，然而，这需要大量的剪枝时间，因此，这里不讨论该问题。



(a)　森林准确率vs.森林成员个数

(b)　森林规模vs.森林成员个数

图 3　在 4 个数据集上的结果

实验 3　评估森林剪枝算法 FTCG 能否提高由 bagging 构建的基于决策树的组合分类器的性能，分别考察了 bagging 构建的决策树是否剪枝两种情况。每个组合分类器都包含 30 棵决策树。表 1 和表 2 给出了 18 个数据集上的实验结果，其中“Bagging”列显示 bagging 建立的组合分类器的分类准确率和标准差，而“FTCG”列是森林剪枝后的对应结果。比较方法采用逐对 t-检验，显著水平为 95%(下同)。从表 1 和表 2 可以看出，不管基分类器是剪枝后的决策树还是未剪枝的决策树，FTCG 显著提升了 bagging 的准确率并降低了它的规模。

表 1　FTCG 和 bagging 的准确率与标准差

Data Set	Unpruned C4.5		Pruned C4.5	
	FTCG	Adaboost	FTCG	Adaboost
australian	87.13±3.38	86.09±3.61•	86.80±3.50	85.86±3.71•
backache	85.06±4.45	83.17±5.99•	85.89±3.48	83.17±5.77•
balance-scale	78.91±3.68	75.07±4.20•	79.79±3.62	76.64±4.29•
breast-cancer	69.94±7.06	67.15±8.47•	69.94±7.12	66.57±8.15•
cars	86.49±4.99	86.78±5.01	86.78±5.44	86.28±5.11
credit-rating	86.48±3.76	85.54±3.78•	86.33±3.52	85.43±3.56•
german-credit	75.29±3.33	73.83±3.82•	74.85±3.08	73.11±3.48•
ecoli	84.46±6.07	83.32±6.61•	84.20±5.41	83.40±6.05•
hayes-roth	78.75±9.93	78.63±9.66	78.75±9.57	76.31±9.16•
heart-c	80.93±6.60	80.34±6.91	81.00±6.06	80.27±6.74
horse-colic	84.51±5.43	83.29±5.24•	84.34±5.51	83.42±5.31•
ionosphere	93.99±4.13	93.93±3.96	93.59±3.97	93.71±3.95
iris	93.53±6.86	94.27±4.83	94.53±5.14	94.53±5.05
lymphography	83.79±8.05	83.43±9.02	84.55±8.59	83.25±8.40•
page-blocks	97.02±0.71	97.04±0.71	97.04±0.75	97.06±0.70
pima	75.09±4.11	74.27±4.31•	75.46±4.00	74.06±3.91•
prnn-fglass	78.14±7.87	78.46±7.88	77.62±8.14	77.84±8.04
vote	95.77±2.93	95.13±3.24•	95.67±2.89	95.33±2.97

注：•(◦)表示 FTCG 显著优于(劣于)bagging，其中，使用显著度为 0.05 的 T 测试检测显著性。

表 2　FTCG 和 bagging 的结点大小与标准差

Data Set	Unpruned C4.5		Pruned C4.5	
	FTCG	Bagging	FTCG	Bagging
australian	4440.82± 223.24	5950.06± 210.53•	2194.71± 99.65	2897.88± 98.86•
backache	1162.79± 96.58	1592.80± 75.97•	518.77± 40.49	764.24± 37.78•
balance-scale	3458.52± 74.55	4620.58± 78.60•	3000.44± 71.76	3762.60± 65.55•
breast-cancer	2164.64± 156.41	3194.20± 144.95•	843.96± 129.44	1189.33± 154.08•
cars	1741.68± 60.59	2092.29± 55.31•	1569.11± 57.55	1834.91± 46.80•
credit-rating	4370.65± 219.27	5940.51± 223.51•	2168.11± 121.51	2904.40± 99.73•
german-credit	9270.75± 197.62	11464.19± 168.63•	4410.11± 114.94	5421.60± 107.24•
ecoli	1366.62± 61.68	1736.42± 64.91•	1304.30± 54.39	1611.02± 56.31•
hayes-roth	498.65± 28.99	697.58± 40.87•	272.30± 45.11	308.48± 53.86•
heart-c	1503.46± 65.47	1946.94± 62.52•	1049.29± 52.06	1317.41± 47.35•
horse-colic	2307.67± 106.99	3625.23± 116.63•	647.89± 102.15	974.93± 129.83•
ionosphere	552.49± 61.41	680.43± 69.95•	521.83± 58.01	634.73± 64.44•
iris	168.46± 111.12	222.66± 150.42•	144.52± 97.26	191.84± 133.12•
lymphography	1089.87± 67.16	1394.37± 61.85•	711.62± 37.61	856.44± 30.83•
page-blocks	1420.05± 278.51	2187.45± 555.02•	1394.11± 225.00	2092.93± 403.79•
pima	2202.41± 674.18	2776.77± 852.95•	2021.19± 600.06	2481.65± 747.19•
prnn-fglass	1219.98± 39.85	1398.62± 36.29•	1145.20± 39.76	1269.28± 35.52•
vote	303.06± 124.00	527.80± 225.05•	174.04± 77.61	276.00± 127.46•

注：•(◦)分别表示 FTCG 的规模显著小于(大于)bagging，其中，使用显著度为 0.05 的 T 测试检测显著性。

实验 4　评估 FTCG 能否有效地提高由基分类器选择方法选择的子组合分类器的性能。先用 bagging 建立包含 200 棵决策树的组合分类器，然后用 EPIC 算法选择其中 30 棵，

最后用我们的 FTCG 进行森林剪枝(用 EPIC-FTCG 表示)。相关的结果(准确率和森林规模)如表 3 所列,其中,左边为准确率比较结果,右边为森林规模的比较。从表 3 可以看出,EPIC-FTCG 显著优于由 EPIC 选择出来的子组合分类器的泛化能力,且降低了组合分类器的规模。

表 3 FTCG 和 EPIC 的准确率(左)与规模(右)以及相应的标准差

Data Set	Unpruned C4.5		Pruned C4.5	
	FTCG	Bagging	FTCG	Bagging
australian	86.83± 3.72	86.22± 3.69•	2447.50± 123.93	3246.16± 116.07•
backache	84.83± 4.46	82.11± 5.89•	708.01± 54.55	931.44± 51.16•
balance-scale	79.74± 3.69	78.57± 3.82•	3277.76± 85.07	4030.82± 94.67•
breast-cancer	70.26± 7.24	67.16± 8.36•	843.96± 129.44	1189.33± 154.08•
cars	87.02± 5.06	86.83± 5.04	1783.32± 60.44	2022.81± 53.19•
credit-rating	86.13± 3.92	85.61± 3.95•	2414.60± 123.66	3226.25± 131.46•
german-credit	74.98± 3.63	73.13± 4.00•	4410.11± 114.94	6007.28± 124.30•
ecoli	83.77± 5.96	83.24± 5.98•	1498.86± 62.27	1806.26± 70.98•
hayes-roth	78.75± 9.57	76.81± 9.16•	275.09± 47.90	311.32± 57.05•
heart-c	81.21± 6.37	79.99± 6.65•	1230.14± 54.80	1510.57± 52.56•
horse-colic	84.53± 5.30	83.80± 6.11•	940.07± 66.64	1337.60± 75.73•
ionosphere	93.90± 4.05	94.02± 3.83	590.63± 65.62	706.79± 73.17•
iris	94.47± 5.11	94.47± 5.02	152.58± 108.04	197.80± 141.31•
lymphography	81.65± 9.45	81.46± 9.39	858.42± 46.50	1022.67± 39.68•
page-blocks	97.02± 0.74	97.07± 0.69	1396.63± 237.03	2086.89± 399.10•
pima	74.92± 3.94	74.03± 3.58•	2391.95± 764.16	2910.31± 936.70•
prnn-fglass	78.13± 8.06	77.99± 8.44	1280.14± 43.85	1410.84± 39.59•
vote	95.70± 2.86	95.33± 2.97	177.36± 86.10	281.62± 140.6•

注:• 表示 FTCG 的性能显著优于(规模显著小于)EPIC,。表示 FTCG 的性能显著劣于(规模显著大于)EPIC,其中,使用显著度为 0.05 的 T 测试检测显著性。

实验 3 和 4 表明:无论森林是基于某种算法(如 bagging)构建的还是某种组合分类器选择算法(如 EPIC)的结果,无论每棵决策树是未剪枝的还是剪枝后的,森林剪枝都能在大部分数据集上显著提高剪枝后的组合分类器的分类准确率。

由于像 EPIC[1] 和 DHCEP[6] 这样的算法可以通过精心选择的子组合分类器提高组合分类器的性能,而森林剪枝又可以进一步提高组合分类器的性能,因此,把它们联合使用将有助于建立更好的组合分类器。

结束语 本文把基于决策树的组合分类器看作森林,提出了一种森林剪枝算法 FTCG。为了确定森林的哪些决策树的哪些分枝可以被剪枝,我们提出一种称作贡献增益的度量,用来评估一个结点生长成一棵子树对组合分类器贡献的提高程度。基于贡献增益,我们设计并实现了一种森林剪枝算法 FTCG,迭代地对森林的每棵决策树进行剪枝,以简化组合分类器,并提高它的分类准确率。

在 18 个数据集上的测试结果表明,无论森林是基于某种算法(如 bagging)构建的还是某种组合分类器选择算法(如 EPIC)的结果,无论每棵决策树是未剪枝的还是剪枝后的,FTCG 都能进一步降低每棵决策树的规模,并且在大部分数据集上显著提高了剪枝后的组合分类器的分类准确率。

森林剪枝的关键是设计一个好的标准,用来评估剪掉一个分枝对组合分类器的影响程度。尽管实验表明本文提出的结点贡献增益较好地度量了结点生长成一棵子树对整个组合分类器分类准确率的提高,但是这方面的研究才刚刚开始,我们期待更好的评估标准的出现。此外,如何把本文提出的贡献度量用于组合分类器的基分类器选择,并与已有的方法进行比较也是值得进一步探讨的问题。

参 考 文 献

[1] Lu Z Y, Wu X D, Zhu X Q, et al. Ensemble Pruning via Individual Contribution Ordering[A]// Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010[C]. 2010:871-880

[2] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2):123-144

[3] Freund Y, Schapire R F. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computer and System Sciences, 1997, 55(1):119-139

[4] Rodriguez J J, Kuncheva L I, Alonso C J. Rotation forest: A new classifier ensemble method[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI), 2006, 28(10):1619-1630

[5] Martinez-Munoz G, Suacutearez A. Pruning in ordered bagging ensembles[C]// Proceedings of the Twenty-Third International Conference on Machine Learning(ICML 2006). 2006:609-616

[6] Partalas I, Tsoumakas G, Vlahavas I P. An ensemble uncertainty aware measure for directed hill climbing ensemble pruning[J]. Machine Learning, 2010, 81(3):267-282

[7] Guo H P, Fan M. Ensemble Pruning via Base-Classifier Replacement[A]// The 12th International Conference on Web-Age Information Management(WAIM)[C]. Springer 2011:505-516

[8] Quinlan J R. C4.5: programs for machine learning[M]. Morgan Kaufmann, 1993

[9] Webb G I. Further Experimental Evidence against the Utility of Occam's Razor[J]. Journal of Artificial Intelligence Research, 1996, 4:397-417

[10] Kuncheva L I. Combining Pattern Classifiers: Methods and Algorithms[J]. John Wiley and Sons, 2004

[11] Schapire R, Freund Y, Bartlett P, et al. Boosting the margin: A new explanation for the effectiveness of voting methods[J]. The Annals of Statistics, 1998:1651-1686

[12] Breiman L. Arcing the edge[R]. University of California, Berkeley, CA, 1997

[13] Asuncion D N A. UCI machine learning repository[OL]. http://archive.ics.uci.edu/ml/, 2007

[14] Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)[M]. Morgan Kaufmann, 2005