计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200600055

基于加权划分非平衡决策树的诗歌朗读情感度分析

董本清　李凤坤
大连东软信息学院　辽宁 大连 116023
(dongbenqing@neusoft.edu.cn)

摘　要　本文面向诗歌朗读的情感度分析,提出了一种新的非平衡决策树算法。该算法称为加权划分非平衡决策树(Weighted Division of Unbalanced Decision Tree,WDOUDT),通过对诗歌朗读感染力的指标展开研究,从朗读的音频中提取梅尔频率倒谱系数,应用可解释性最强的决策树方法进行建模。加权划分非平衡决策树推导算法不使用进化算法和启发信息搜索,应用在诗歌朗读音频的情感度打分中,时间复杂度低于传统决策树,该算法具有更少的节点数和较好的泛化能力,对噪音数据有较好的鲁棒性。

关键词:非平衡决策树;加权划分;情感度分析;快速收敛
中图法分类号　TP391.43

Analysis of Emotional Degree of Poetry Reading Based on WDOUDT

DONG Ben-qing and LI Feng-kun
Dalian Neusoft University of Information,Dalian,Liaoning 116023,China

Abstract　In this paper,a new unbalanced decision tree algorithm for infectious expressions of reading poem is proposed. This algorithm called Weighted Division of Unbalanced Decision Tree (WDOUDT). Through the study on the index of poetry reading appeal,mel-frequency cepstral coefficients are extracted from the reading audio,and the decision tree method with the strongest interpretability is used for modelling. WDOUDT does not use evolutionary algorithm and heuristic information search,it is applied to the emotional scoring of poetry reading audio,and the time complexity is lower than the traditional decision tree. The proposed algorithm has fewer nodes and better generalization ability,and has better robustness to noise data.

Keywords　Unbalanced decision tree,Weighted division,Infectious expression,Fast convergence.

1　引言

自然语言处理是人工智能的一个重要研究领域,目前人们对自然语言的各个方面都展开了深入研究,如机器翻译、内容生成、摘要抽取、文本语音互转、情感识别等,取得的成果已经远远超过了以往的任何时候,并且有些领域的成果已经可以应用到产业界,例如百度的机器翻译,科大讯飞的语音识别等。

在人类的自然语言中,诗歌是一类非常特殊的体裁,有独特的节奏和韵律,朗读的时候情感要比其他体裁的文章更丰富。目前对于朗读的感情的识别,主要存在以下两种方法:一是提取连续语音在不同情感度朗读模式下的基频曲线进行定量分析与比较。二是通过构造适应用于语音情感识别的卷积神经网络模型来实现情感度识别[1]。但这类方法对语音情感数据库的依赖较大,且情感与文化具有强关联性,不同人有不同的讲话习惯等,这些因素都增加了识别的困难。

决策树是一种常见的归纳式学习方法,可用于分类和回归问题,是目前研究热点[1-2]。构建最优决策树的过程已经被证明是 NP 完全问题[3],为了避免陷入局部最优解,一些研究者采用进化类算法来推导决策树[4-6],但基计算时间复杂度过高,不适用于规模较大的决策树。实际应用中,大部分决策树采用贪心算法进行构建,即采用自顶向下、递归划分的方式进行推导,其中以 Quinlan 提出的 ID3[7] 以及 Breiman 等提出的 CART[8] 最为出名。这些算法使用某种类别不纯度的度量作为启发信息进行搜索,寻找能使纯度下降最大的属性以及阈值。例如:ID3 使用信息增益,C4.5 使用信息增益率,而 CART 使用基尼指数等。

更多的决策树构建方法在近几十年被提出,这些方法中大部分都对非叶子结点采用线性划分:在属性空间中,选择一个或多个超平面将属性空间划分为两个或多个区域,非叶结点中的样本根据所处的区域被划分,从而形成下一层的结点。Ho[9] 根据超平面的方向、数量以及选取方法,将划分方法归结为 3 类,即轴平行线性划分、斜划分和分段划分。

轴平行划分是最常见的划分方式,其产生的决策树属于单变量决策树,在选择属性为数值属性时,需要测试 $x_i>\theta$,这里 x_i 是一个样本的属性值,θ 是常数,也称为划分点 cut。每次对一个结点进行划分时,需要根据预设的划分准则(信息增益、gini 等),在全部属性中选择局部最优的属性和划分点。由于进行了启发搜索,这种推演方式具有很快的速度,划分结

基金项目:辽宁省博士启动基金(20170520398);辽宁省教育厅科学技术一般项目(L2015041)
This work was supported by the Liaoning Provincial Doctor Start-up Fund (20170520398) and General Project of Science and Technology of Liaoning Provincial Department of Education (L2015041).
通信作者:李凤坤(lifengkun@neusoft.edu.cn)

点的平均时间复杂度为 $O(d*n*\log n)$，其中 d 是属性个数，n 是样本个数。但是，每一步搜索得到的是局部最优解，而非全局最优解。这种方式产生的树往往比较深。

斜划分产生的是多变量决策树，它使用多个属性的线性组合，需要对样本进行如下测试：

$$\sum^{d} a_i x_i + a_{d+1} > 0 \tag{1}$$

其中，a_i 表示系数，x_i 表示样本在第 i 个属性上的值。显然，这类方法产生的划分超平面是斜的，不一定与某个属性轴平行，系数 a_i 决定了超平面的方向和位置（截距）。通常情况下，寻找合适的斜划分超平面要比轴平行划分超平面复杂，并且需要付出更大的计算量，但得到的树往往比较浅，树中总的结点数更少且泛化能力更好，但会损失一些可理解性。在此基础上，更多寻找斜树划分的方法被提出，其中一些采用了优化搜索或启发策略寻找局部最优划分，例如 SADT[10] 使用模拟退火算法，OC1[11] 组合了 CART-LC 和 SADT 中的策略，另一些则研究使用线性判别分析 LDA、线性回归、SVM 等统计方法来确定划分平面，如 quest 使用线性判别分析（LDA）[12]，而 Mui 等[13] 采用 FDA 等。

W-k-MEANS[14] 算法对非叶子结点上的数据集分簇，从而得到决策模型。实验表明，该方法不仅具有较高的时间效率，而且获得了较高的正确率。受该想法的启发，本文提出了加权划分非平衡决策树推导算法，不使用进化算法和启发信息搜索，因此计算速度更快，产生的决策树具有与 C4.5，CART 等经典算法相近或更好的泛化能力和可解释性。

2 知识准备

本文提出的决策树推导算法需要使用 relief_f 算法以及 k-means，kmodes 和 KPrototype 算法，为了更好地阐述本文算法，本章先简要回顾一下这些算法。

2.1 relief 和 relief_f 算法

属性也被称为特征，对于一个分类任务来说，给定属性集，有些属性可能很有用，称为相关特征；有些属性没有用，称为无关特征。Relief 是非常著名的过滤式特征选择算法，由 Kira 等[15] 于 1992 年提出。该算法根据各个属性和类别的相关性赋予属性不同的权重，并从训练集 D 中随机选择一个样本 R，然后从和 R 同类的样本中寻找最近邻样本 H，称为 Near Hit，从和 R 不同类的样本中寻找最近邻样本 M，称为 Near Miss，然后根据以下规则更新每个特征的权重：如果 R 和 Near Hit 在某个特征上的距离小于 R 和 Near Miss 上的距离，说明该特征对区分同类和不同类的最近邻是有益的，则增加该特征的权重；反之，如果 R 和 Near Hit 在某个特征上的距离大于 R 和 Near Miss 上的距离，说明该特征对区分同类和不同类的最近邻起负面作用，则降低该特征的权重。重复以上过程 m 次，最后得到各特征的平均权重。特征的权重越大，表示该特征的分类能力越强；反之，表示该特征分类能力越弱。

$$W(A) = W(A) - \frac{diff(A,R,H)}{m} + \frac{diff(A,R,M)}{m} \tag{2}$$

其中，$W(A)$ 为属性 A 的权重，R 表示随机抽取的样本，H 表示 R 的猜中最近邻，M 表示 R 的猜错最近邻，$diff(\cdot)$ 是用于计算两个样本在属性 A 上的距离的函数。当 A 为连续属性时，$diff$ 一般为规范化到 $[0,1]$ 区间的曼哈顿距离；当 A 为离散属性时，两个样本属性值相同，$diff=0$，否则 $diff=1$，即：

$$diff(A,R_1,R_2) = \frac{|R_1(A) - R_2(A)|}{\max(A) - \min(A)} \tag{3}$$

Relief 算法只适用于二分类问题，1994 年 Kononenko 提出了 Relief_f[16] 算法，并将其扩展到多分类问题。权重按照如下公式计算：

$$W(A) = W(A) - \frac{diff(A,R,H)}{m} + \sum_{C \neq class(R)} \frac{P(C) \times diff(A,R,M(C))}{m} \tag{4}$$

其中，C 表示与抽样样本 R 所属类不同的其他类别，$P(C)$ 表示类别 C 的样本在所有与 R 异类的样本中所占的比例。

Relief 和 *Relief_f* 算法的运行时间随着样本的抽样次数 m 和原始特征个数 N 的增加而线性增加，因而运行效率非常高，可用于大规模数据集。本文使用 *Relief_f* 算法为属性加权，结点划分计算距离时，使用带有权重的距离代替实际距离。

2.2 Kmeans，KMode，KPrototype

Kmeans 聚类是一种动态聚类算法，基于聚类准则函数最小化的原则，通过迭代将数据划分到不同的类中，使生成的类尽可能的紧凑和独立。其算法思想为：设输入样本为 $S=x_1,x_2,\cdots,x_n$，假设聚类数为 k，每一个簇 $C=\{C_1,C_2,\cdots,C_k\}$ 的中心代表一类数据集簇。首先选择初始的 k 个簇中心 $\mu_1,\mu_2,\cdots,\mu_k$；然后对于每个样本 x_i，将其标记为距离簇中心最近的簇，即：

$$label_i = \underset{1 \leqslant j \leqslant k}{\operatorname{argmin}} \| x_i - \mu_j \| \tag{5}$$

最后将每个簇 C_j 中心更新为隶属为该簇的所有样本的均值，计算新的簇中心点 μ_j。

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i, x_i \in C_j \tag{6}$$

直到聚类准则函数收敛，其函数为：

$$E = \sum_{j=1}^{k} \sum_{i=1}^{N_j} \| x_i - \mu_j \|^2, x_i \in C_j \tag{7}$$

E 为 kmeans 算法聚类所得的簇 C_j 划分的最小化平方误差，E 值越小，簇内样本相似度越高。该算法的时间复杂度为 $O(I*n*k)$，其中 I 为迭代次数，n 为数据量。

Kmodes 算法是 Kmeans 算法在离散属性数据集上的变体，计算簇中心时，使用众数代替式(4)中的均值；计算样本与中心的距离时，距离为计算样本与簇中心相异属性的个数。

KPrototype 是 KMeans 与 Kmodes 的结合，用于同时存在连续属性和离散属性的数据集，计算样本和簇中心距离时，采用式(8)：

$$dist = (1-\alpha) \times dist_N + \alpha \times dist_C \tag{8}$$

$dist\_N$ 和 $dist\_C$ 分别表示属性集中连续属性子集产生的距离和离散属性子集产生的距离，α 为系数。

本文在面对连续属性数据集、离散属性数据集以及混合属性数据集时，分别使用 Kmeans，KMode 和 KPrototype 算法产生的聚类中心代替聚类中心（centroid of class）作为锚点，对数据进行划分。下一节将阐述这样做的理由和具体做法。

3 加权划分非平衡中决策树算法

3.1 基础算法 basic

本文提出的加权划分非平衡决策树算法是根据样本到锚

点的距离来对锚点进行划分的，首先给出基础算法 **basic**，其中锚点选择的是训练集中各个类别样本的质心。

算法 1 basic 算法

输入：训练集 D，with n 个样本，k 个类别，d 个属性。minparent，mingini

$D=\{x_1 y_1, x_2 y_2, \cdots, x_n y_n\}$

$A=\{a_1, a_2, \cdots, a_d\}$

$L=\{l_1, l_2, \cdots, l_k\}$

$X_i=\{x_{i1}, x_{i2}, \cdots, x_{id}\}$

Y_i 属于 L

输出：以 node 为根的决策树 T

过程：函数 TreeGenerate(D)

1. 生成结点 node
2. if D 中样本数量小于阈值 minparent 或 D 中样本 gini 指数小于等于阈值 mingini 或样本集不可分（即样本集中存在矛盾数据，所有属性值相同但类别不同） Then
3. 将 node 标记为叶结点，其类别标记为 D 中样本数最多的类。return
4. end if
5. 根据当前样本集合类别数量 k，定义 k 个 d 维度向量 $c_1, c_2, \cdots, c_k$，用于保存 k 个类别的中心
6. for i=1 to k do
7. for j=1 to d do
8. c_{ij}=center(c_i, a_j)
9. end for
10. end for
11. 定义 k 个子集合 $D_1, \cdots, D_k$，全部初始为空集
12. for i=1 to n do
13. min=select_near(xi,C)
14. 将 x_i 加入集合 Dmin
15. end for
16. for i=1 to k do
17. for j=1 to d do
18. c_{ij}=center(D_i, a_j)
19. end for
20. end for
21. for i=1 to k do
22. if(D_i 非空)
23. 将 TreeGenerate(D_i)返回的子树根结点作为当前结点 node 的分支结点
24. end if
25. end for
26. return node

算法 basic 中，产生多叉树，每个非叶节点的分支数量与该节点内样本类别数量相同。算法的输入除了包含训练集 D 之外，还包括两个用于预剪枝的阈值 minparent 和 mingini，分别表示非叶结点的最少样本数和最小 gini 指数。算法开始时，如果当前结点不满足划分的条件，则将当前结点标记为叶子结点，其类别标记为结点中样本数最大的类（2－4）。如果需要对结点进行划分，首先，计算当前样本集合各个类别的质心（5－10）作为锚点，计算质心使用函数 *center*(c_i, a_j)。如果属性 a_j 是数值属性，则 *center* 得到的是类别为 c_i 的样本的均值，当属性 a_j 为分类属性，则 *center* 得到的是类别为 c_i 的样本的众数。然后，为 D 中的每个样本 x_i 计算离它最近的锚点，并将样本分配到相应子集合（11－15），为样本 x_i 选择最近锚点的函数为 *select_near*(x_i, C)。在完成样本划分后，需要再次调用函数 *center*(D_i, a_j)重新计算各个子集 D_i 的中心（16－20），这些中心需要保存在树模型中，在测试看不见的数据时使用。最后，使用每个非空子集 D_i 作为输入递归调用函数 *TreeCenerate*，产生下一级划分（20－22），并返回根结点。

算法 basic 采用最近聚类中心（centroid of class）作为划分准则，虽然能够快速生成决策树，但经过实验发现，在本研究中所面临的经处理的音频数据集上，该方法生成的决策树往往比较大，且泛化能力差，主要归因于以下几个方面：

（1）算法 basic 没有采用样本集不纯度等指标进行启发搜索，划分依据指标不清晰；

（2）没有考虑不相关、弱相关属性对计算距离产生的坏的影响；

（3）未考虑样本的空间分布，计算得到的聚类中心有时并不能很好地代表相应类别的空间分布。

因此，以算法 basic 为基础，进行了两方面的改进，3.2－3.3 节将进行详细描述。

3.2 为属性加权

经典的决策树算法，如 C4.5 和 CART 等，被认为是自带属性筛选能力的算法，其在全部属性中选择最有利于不纯度下降的属性对结点进行划分，因此这些算法能够有效地对付不相关属性和冗余属性。但算法 basic 在计算样本到质心的距离时，平等地看待全部属性，当存在不相关属性时，必然会减弱“样本到聚类中心距离的大小”与“分类问题本身”的相关性。

于是很自然地想到，如果能够为属性加权，则与分类问题相关的属性就拥有较大的权重，反之，拥有较小的权重，这样在计算距离时使用权重放大强相关属性对距离的影响，减小弱相关和不相关属性对距离的影响，可以有效提高树的生成效率和正确率。

所提算法的第一个改进是：在结点划分之前，使用 relief_f 算法和当前结点中的样本集来计算每个属性的权重。图 1 使用一个小例子说明了属性加权的意义所在。图 1 中，矩形代表类别 1、圆形代表类别 2，两个加号分别代表两个类别的质心，实线代表属性未加权产生的分割线，虚线代表 x 轴属性权重为 0.2、y 轴属性权重为 0.8 时产生的分割线，显然纠正了类别 2 样本的划分。

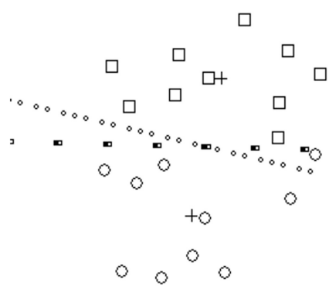

图 1 加权的作用示意图

Fig. 1 Schematic diagram of weights effection

为了进一步说明属性加权的作用，我们使用真实世界的数据集 iris 进行了简单的实验，iris 数据集的 150 个样本来自 Iris-setosa，Iris-versicolor，Iris-virginica 3 个类别，每个类别 50 个样本。使用算法 basic 中描述的划分方法对数据集进行一次划分，得到的结果是：

子节点 1：Iris-setosa 50

子节点 2：Iris-versicolor 44，Iris-virginica 4

子节点 3：Iris-versicolor 6，Iris-virginica 46

错分样本数量为 10 个。而使用随机分层 10％采样调用 relief 算法为 4 个属性分别得到权重 0.090 740 7，0.138 889，0.342 938，0.386 111 后，通过加权计算得到的距离进行 1 次划分，得到的结果是：

子节点 1：Iris-setosa 50

子节点 2：Iris-versicolor 48、Iris-virginica 4

子节点 3：Iris-versicolor 2、Iris-virginica 46

错分样本数量为 6 个，如果此时停止划分，使用连根结点在内的 4 个结点组成的树作为模型，在训练集上已经能够达到 96％的正确率。

采用为属性加权方式对算法 basic 的改进称为算法 W。

3.3 使用聚类算法产生的簇的中心作为锚点

当数据集的某些类的样本分布在属性空间的不同区域时，类别的质心不能够很好地代表该类别样本，如图 2 所示，矩形类别分布在两个不同的区域，若使用该类别的质心和圆形聚类中心连线上的中垂线分割样本，效果显然不好。然而，图 2 中的样本显然形成了两个簇，如果使用两个簇的中心连线的中垂线对样本集进行划分，至少可以将图左侧的 5 个矩形类别的样本分出来。

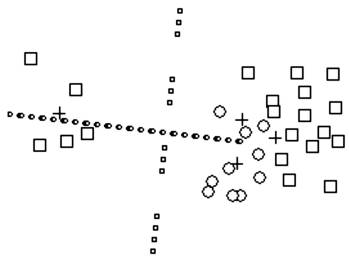

图 2 聚簇中心代替质心作为锚点的效果示意图

Fig. 2 Schematic diagram of cluster center replacing center of mass as anchor point

本文提出的第二个改进是在算法 basic 的基础上，使用 kmeans 算法产生的簇中心代替聚类中心作为锚点，对样本进行划分。也可以理解为，kmeans 产生的簇就是当前结点样本集合的划分结果。具体做法是：将类型质心作为 kmeans 算法的初始簇中心，调用 kmeans 算法，待 kmeans 算法收敛或达到预设最大迭代轮次数（本文默认为 6 次）后，将最后一轮产生的簇作为当前结点的划分结果（各个子集）。实际上，这一改进仅仅是将算法 basic 中第 11－20 步的操作重复了若干次。我们称之为算法 C。

当全部属性都是类别属性时，kmeans 算法变换为 kmode 算法，当属性集既包含数值属性又包含类别属性时，使用 KPrototype 算法。

另外，可以考虑将第一点改进算法 W 和第二点改进算法 C 相结合，形成算法 WC。

3.4 改进后的算法 WDOUDT

没有一个算法适用于所有的数据集，同样，本文提出的 2 个改进也未必适合全部的情况。面对这种情况，我们在提出的算法中增加了一个整型参数 p，取值 0～3。p 为 0 时，表示使用算法 basic；p 包含 1 时，表示加入改进 1；p 包含 2 时，表示加入改进 2。具体使用哪种算法由用户自行实验确定。

经过上述改进之后的算法称为加权划分非平衡决策树，具体如下：

输入：训练集 D，with n 个样本，k 个类别，d 个属性，minparent，mingini，参数 p

D＝{$x_1 y_1$，$x_2 y_2$，…，$x_n y_n$}

A＝{a_1，a_2，…，a_d}

L＝{l_1，l_2，…，l_k}

X_i＝{x_{i1}，x_{i2}，…，x_{id}}

Y_i 属于 L

输出：以 node 为根的决策树 T

过程：函数 TreeGenerate(D)

```
1. 生成结点 node
2. if D 中样本数量小于阈值 minparent 或 D 中样本 gini 指数小于等于阈值 mingini 或样本集不可分（即样本集中存在矛盾数据，所有属性值相同但类别不同） Then
3.     将 node 标记为叶结点，其类别标记为 D 中样本数最多的类。return node
4. end if
5. 根据当前样本集合类别数量 k，定义 k 个 d 维度向量 c1，c2…ck，用于保存 k 个类别的中心
6. for i＝1 to k do
7.     for j＝1 to d do
8.         c_ij＝center(c_i，a_j)
9.     end for
10. end for
11. 定义权重向量 w[k]，初始为全 1。
12. if p 包含 1 then
13.     调用 relief_f 算法，更新 w[k]
14. end if
15. 定义 k 个子集合 D1…Dk
16. while(迭代次数小于 6) do
17.     将 D1. Dk 清空
18.     for i＝1 to n do
19.         if p 包含 4 then
20.             min＝select_near_unxxx(x_i，C，N，W) //N 表示各类别样本数量的向量
21.             else
22.                 min＝select_near(x_i，C，W)
23.             end if
24.             将 x_i 加入集合 D_min
25.     end for
26.     for i＝1 to k do
27.         for j＝1 to d do
28.             c_ij＝center(D_i，a_j)
29.         end for
30.     end for
31.     if(p 不包含 2 或 c_ij与之前一轮 c_ij无变化) then
32.         break while
33.     end if
34. end while
35. for i＝1 to k do
36.     if(Di 非空)
37.         将 TreeGenerate(D_i)返回的子树根结点作为当前结点 node 的分支结点
38.     end if
```

39. end for
40. return node

算法 WDOUDT 相比算法 basic 增加了参数 p，用于决定应用哪些改进。算法第 11－14 行，在 p 包含 1 时，需要调用 relief_f 算法计算各个属性的权重，如果不需要，则各个属性的权重均为 1；算法第 16－34 行，是使用各个聚类中心作为初始簇中心的 k_means 算法，k_means 算法将样本聚为 k 类，k 是该结点样本集合中类别的数量，如果参数 p 不包含 2，即不使用改进 2 时，while 仅迭代 1 轮后退出，否则需要等到 k_means 算法收敛或达到最大迭代轮次数。

4 时间复杂度分析

经典的轴平行划分结点算法 C4.5，CART 等，划分结点时，需要在全部 d 个属性中寻找能够使不纯度下降最大的那个属性，当面对的属性为数值属性时，还要找到最佳的划分点，为了得到这个划分点，需要为全部 n 个样本按照该属性值排序，然后在出现类别改变的划分点上选择最佳划分点。因此，这类算法为完成一个结点划分的时间度约为 $O(dn\log n)$，d 为属性数量，n 为样本数量。

本文提出的算法 basic 划分结点分为 3 个阶段，阶段 1 是计算各个类别的质心，完成该计算需要遍历 n 个样本的 d 个属性，因此时间复杂度为 $O(dn)$，阶段 2 是为全部样本选择最近锚点，需要计算每个样本与每个锚点的距离，时间复杂度为 $O(kdn)$，k 为锚点个数，即类别数量；阶段 3 是计算划分后各个样本子集的中心，时间复杂度为 $O(dn)$，因此算法 basic 的时间复杂度为 $O(kdn)$。

算法 WDOUDT 在参数 p 为 3 时，应用全部 2 个改进，改进 1 需要使用 relief_f 计算属性权重，时间复杂度 $O(dmn)$，d 是属性个数，n 是样本数，m 是采样样本数，通常 m 远远小于 n，便可得到可靠权重，本文实现的 relief_f 算法在样本集中类别样本数较小时（<500），采样数为该类样本的 10%，在该类样本数较大时（≥500），该类采样 50 个。改进 2 相对于算法 basic 仅仅是重复算法 basic 的阶段 2 和 3 I 次，本文中 I 不超过 6。

综上，算法 WDOUDT 划分一个结点时的时间复杂度在最坏的情况下为 $O(mdn+Ikdn)$，$O(dn*(m+ik))$，由于 m（relief_f 算法采样数）和 I（kmeans 算法迭代次数）、k（类别数）相对 n（样本数）较小，因此认为算法 WDOUDT 与经典轴平行算法的时间复杂度相近。另外，前文比较的是对于一个结点的划分的时间复杂度，而不是整棵树的推导的时间复杂度。经典轴平行算法使用数值属性对结点进行划分时，进行的是 2 路划分，而本文算法 WDOUDT 进行的是 k 路划分，这相当于进行多次 2 划分。经典轴平行算法产生的树更窄、更深。本文算法产生的树更宽、更深，在大部分情况下，本文算法的划分次数少于经典轴平行算法，因此认为在树结点数量相近的前提下，本文算法比经典轴平行算法的时间效率更高。

5 实验

5.1 数据预处理

本文与某机构合作，采集到在校小学生诗歌朗读音频数据 800 份，由 5 名小学教师和广播员对音频的情感度进行打分，最高得分为 5 分，最低得分为 1 分。然后将 5 个人对每个音频的平均得分进行四舍五入，作为音频的最终得分。

对于每个音频，将每一帧数据采用梅尔频率倒谱系数（mel-frequency cepstral coefficients，MFCC）进行计算，但是只有 MFCC 被用作特征，特征的数目相对于样本的数目来说太少了。因此，又继续计算了最小值、最大值、平均值、中位数、1/4 分位数、3/4 分位数、标准差等指标，所有这些都作为特征。

经过这种处理，将连续的音频信号转换为本文算法可以处理的数值数据。

5.2 准确率与时间的关系

对于单个决策树，采用 WDOUDT，C4.5 和 CART 算法进行比较，同样使用 10 折交叉验证方式，在 WDOUDT 中，决策树深度大于 5，准确率就可达到稳定；而在 C4.5 和 CART 中，决策树深度大于 8 时，准确率才达到稳定，准确率和时间的关系如表 1 所列。

表 1 3 种算法的准确率与时长的比较

Table 1 Accuracy and time length comparison of three algorithms

Algorithm	*Accuracy*/%	Depth of Tree	*Time Length*/h
WDOUDT	78.1	5	2.3
C4.5	75.3	8	4.1
CART	76.1	8	3.8

5.3 生成随机森林的代价

与其他决策树算法一样，本文提出的 WDOUDT 算法对转换之后的音频数据的判别准确率并不是很高，远远达不到产品化的标准，通常都会通过 bagging 和 boosting 来提高准确率，而随机森林是最常使用的一种办法。接下来使用 3 种不同的算法构造决策树并生成随机森林。准确度与决策树数量之间的关系如图 3 所示。

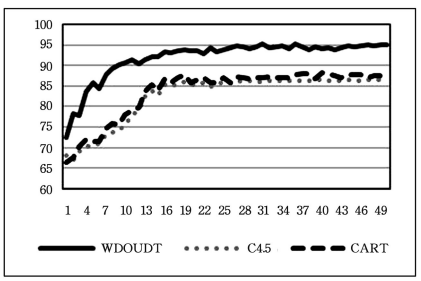

图 3 准确度与决策树数量之间的关系

Fig. 3 Relationship between accuracy and the number of decision trees

接下来，在如下的实验环境中：CPU Intel Core i7-6500U，2 核，每核 2.5 GHz，RAM 16 GB，又测试了不同算法达到相同精度所需要的时间，如图 4 所示。

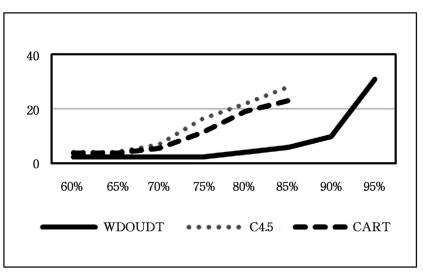

图 4 达到相同准确度所需时间对比

Fig. 4 Comparison of time required to achieve the same accuracy

从以上结果来看，WDOUDT 算法相比 C4.5 和 CART 算

法能得到更好的结果,并且生成单个决策树或随机森林所需的时间更少。然而,3 种算法的精确度到都不是特别高。究其原因,3 种算法的分类误差数据分析主要集中在 1 级和 5 级分类中。由于原始数据中 1 分和 5 分的得分很少(特别是 1 分数据,只有不到 10 项),它们的共同特点不那么统一和明显,而 3 分和 4 分的准确率几乎为 100%,原因是其原始数据最多,样本最为丰富。

结束语 本文提出了一种加权划分非平衡决策树(WDOUDT)的算法,能够有效应用于诗歌朗读情感度识别领域。WDOUDT 基于决策树提出了 2 点改进:1)为属性加权;2)用聚类中心代替质心作为锚点。

实验表明,WDOUDT 比 C4.5,CART 等经典决策树具有更好的精度,适用于大数据量、多样本的情况。

参 考 文 献

[1] NIU Y F. Research on Speech Emotion Recognition Based on Deep Learning[D]. Chongqing:Chongqing University,2018.

[2] GUTIERREZ-RODRÍGUEZ A E,MARTÍNEZ-TRINIDAD J F,GARCÍA-BORROTO M,et al. Mining patterns for clustering on numerical datasets using unsupervised decision trees[J]. Knowledge-Based Systems,2015,82:70-79.

[3] HYAFIL L,RIVEST R L. Constructing optimal binary decision trees is np-complete[J]. Inf. Process. Lett. ,1976,5(1):15-17.

[4] BASGALUPP,MÁRCIO P,BARROS R C,et al. LEGAL-tree:a lexicographic multi-objective genetic algorithm for decision tree induction[C]// Acm Symposium on Applied Computing. ACM,2009.

[5] BASGALUPP M P,DE CARVALHO A C P L F,BARROS R C,et al. Lexicographic multi-objective evolutionary induction of decision trees[J]. International Journal of Bio-Inspired Computation,2009,1(1/2):105-117.

[6] BARROS R C,BASGALUPP M P,FREITAS A A. Automatic Design of Decision-Tree Algorithms with Evolutionary Algorithms[J]. Evolutionary Computation,2013,21(4):659-684.

[7] QUINLAN J R. Induction of Decision Trees[J]. Machine Learning,1986,1(1):81-106.

[8] BUNTINE W L. Learning classification trees[J]. Statistics and Computing,1992,2(2):63-73.

[9] HO T K. The random subspace method for constructing decision forests[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1998,20(8):844.

[10] HEATH D,KASIF S,SALZBERG S. Induction of Oblique Decision Trees[J]. Journal of Artificial Intelligence Research,1993,2:1-32.

[11] MURTHY S K,KASIF S,SALZBERG S. A System for Induction of Oblique Decision Trees[J]. Journal of Artificial Intelligence Research,1996,2(1):1-32.

[12] LOH W Y,SHIH Y S. Split Selection Methods for Classification Trees[J]. Statistica Sinica,1999,7(4):815-840.

[13] MUI J K,FU K S. Automated classification of nucleated blood cells using a binary tree classifier[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1980,PAMI-2(5):429-443.

[14] HUANG J,NG M,RONG H,et al. Automated Variable Weighting in k-Means Type Clustering[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2005,27(5):657-668.

[15] KIRA K,RENDELL L A. The Feature Selection Problem:Traditional Methods and a New Algorithm[C]// Proceedings of the 10th National Conference on Artificial Intelligence. San Jose,CA,AAAI Press,1992:12-16.

[16] KONONENKO I. Estimating attributes:Analysis and extensions of RELIEF[M]// Machine Learning:ECML-94. Springer Berlin Heidelberg,1994.

DONG Ben-qing, born in 1981, Ph.D, associate professor, is a member of CCF. His main research interests include software application and computer education.

LI Feng-kun, born in 1983, master. Her main research interests include intelligent algorithm and artificial intelligence.