

计算机应用研究 优先出版

原创性 时效性 就是科研成果的生命力
《计算机应用研究》编辑部致力于高效的编排
为的就是将您的成果以最快的速度
呈现于世

* 数字优先出版可将您的文章提前 8~10 个月发布于中国知网和万方数据等在线平台

基于动态集成决策树的多类别时间序列分类模型

作者	徐雷, Geoff I Webb, Francois Petitjean, 张阳
机构	西北农林科技大学 信息工程学院 ; 莫纳什大学 信息技术学院
发表期刊	《计算机应用研究》
预排期卷	2018 年第 35 卷第 6 期
访问地址	http://www.arocmag.com/article/02-2018-06-009.html
发布日期	2017-06-14 11:53:09
引用格式	徐雷, Geoff I Webb, Francois Petitjean, 张阳. 基于动态集成决策树的多类别时间序列分类模型[J/OL]. [2017-06-14]. http://www.arocmag.com/article/02-2018-06-009.html .
摘要	针对以往时间序列分类技术忽略了数据间自相关性对算法影响的不足, 通过对传统决策树算法进行扩展, 提出了序列熵和序列对信息增益的概念, 并以此构建针对时间序列的决策树 (time series decision tree, TSDDT)。在此基础上, 以 TSDDT 为基分类器, 通过动态分类器集成技术, 提出了时间序列动态集成分类算法 (En-TSDDT)。在 UCR 数据集上的实验表明, 与目前应用最广泛的 1NN-DTW 分类器相比, En-TSDDT 克服了时间序列数据的自相关性对分类算法的影响, 具有更好的分类性能和鲁棒性。
关键词	时间序列分类, 自相关性, 熵, 决策树, 集成分类器
中图分类号	TP301.6
基金项目	

基于动态集成决策树的多类别时间序列分类模型

徐 雷¹, Geoff I Webb², Francois Petitjean², 张 阳^{1*}

(1. 西北农林科技大学 信息工程学院, 陕西 杨凌 712100; 2. 莫纳什大学 信息技术学院, 澳大利亚 墨尔本 3800)

摘 要: 针对以往时间序列分类技术忽略了数据间自相关性对算法影响的不足, 通过对传统决策树算法进行扩展, 提出了序列熵和序列对信息增益的概念, 并以此构建针对时间序列的决策树 (time series decision tree, TSDDT)。在此基础上, 以 TSDDT 为基分类器, 通过动态分类器集成技术, 提出了时间序列动态集成分类算法 (En-TSDDT)。在 UCR 数据集上的实验表明, 与目前应用最广泛的 1NN-DTW 分类器相比, En-TSDDT 克服了时间序列数据的自相关性对分类算法的影响, 具有更好的分类性能和鲁棒性。

关键词: 时间序列分类; 自相关性; 熵; 决策树; 集成分类器

中图分类号: TP301.6

Multiclass time series decision tree model based on dynamic classifier ensemble

Xu Lei¹, Geoff I Webb², Francois Petitjean², Zhang Yang^{1*}

(1. College of Information Engineering, Northwest A&F University, Yang Ling Shaanxi 712100, China; 2. Faculty of Information Technology, Monash University, Melbourne Victoria 3800, Australia)

Abstract: Many algorithms have been proposed for the problem of time series classification. However, the influence of autocorrelation between series data is always ignored in the current methods. To deal with it, this paper presents a novel decision tree called time series decision tree (TSDDT), which is based on the concept of series entropy and series pairs information gain. Moreover, we propose an Ensemble Classifier (En-TSDDT) based on TSDDT and Dynamic Ensemble methodology. Finally, experiments are conducted on the UCR datasets. The results shows that En-TSDDT out-performs 1NN-DTW in both robustness and accuracy.

Key Words: time series classification; autocorrelation; entropy; decision tree; classifier ensemble

0 引言

时间序列是一类常见且与时间相关的数据, 广泛存在于商业、医学、科学等领域^[1], 具有数据维度高、实值有序、数据间存在自相关性等特点^[2], 即同长度的时间序列中包含有效信息的子序列长度不同^[3]。随着信息技术的发展, 产生的时间序列数据规模越来越大, 决策树技术依靠其对于大数据处理的优势和决策规则的易解释性, 成为人们关注的重点^[4-8]。

传统决策树将序列每一个维度作为一个属性, 通过计算不同属性的信息增益 (Information Gain) 或基尼不纯度 (Gini Impurity) 选择最优属性作为节点分裂属性^[9]。在基于决策树的时间序列分类研究中, 处理节点分裂时均采用传统决策树构建方法, 将时间序列的每一个时刻作为一个属性, 按照时间轴一一对应, 忽略了时间序列的自相关性对某一个属性的影响, 即某个属性的取值与其前后相邻的若干个属性的取值相关。如图 1 所示, 在 50words 数据集中, 上面两个序列为类别 30, 下面

两个为类别 50。两个不同的单词序列存在明显不同, 同时, 同一个单词类别的序列也存在差异。若直接使用传统决策树构建算法, 可能会导致错误的结果。

针对传统决策树算法没有考虑到时间序列自相关性和不对齐性对算法影响的不足, 本文将一个完整的时间序列作为一个属性, 并在传统决策树的属性选择和属性值划分两个方面进行扩展。在属性选择上, 本文以序列对的维数作为特征属性, 代替了传统决策树以序列维数作为特征属性, 大大减小了特征属性的数量, 并提出序列熵的概念以代替传统决策树中所使用的信息熵, 作为属性选择标准。在属性值划分上, 将序列对作为决策树分裂属性, 基于动态时间规整距离 (DTW) 对时间序列集合进行划分, 提出了基于决策树的时间序列分类算法 TSDDT。由于分类器集成技术可以进一步提高分类性能^[10], 在此基础上, 使用动态集成分类技术构建了集成分类器 En-TSDDT。

大量研究表明, 在时间序列分类问题上, 基于动态时间规整的最近邻分类器 1NN-DTW, 与其他传统分类算法相比, 有最

作者简介: 徐雷 (1991-), 男, 陕西高陵人, 硕士, 主要研究方向为机器学习、数据挖掘 (xulei1234@nwfau.edu.cn); GeoffIWebb (1960-), 男, 澳大利亚人, 教授, 博士, 主要研究方向为人工智能、机器学习、大数据等; FrancoisPetitjean (1987-), 男, 法国人, 高级讲师, 博士, 主要研究方向为数据挖掘、机器学习、图像处理; 张阳 (1975-), 男 (通信作者), 教授, 博士, 主要研究方向为数据挖掘、机器学习。

好的分类性能^[8, 11]。

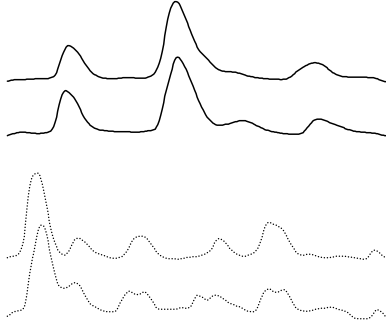


图 1 50words 数据集中两个不同单词的不同序列

在 UCR 数据集上的实验表明，与目前分类性能最好的 INN-DTW 相比，En-TSDT 具有更强的鲁棒性，更高的准确率及 F1 值。

1 时间序列决策树及时间序列集成分类器

首先引入时间序列数据分类的问题定义，然后针对传统决策树信息熵和信息增益，扩展出序列熵及序列对信息增益，给出其定义及计算方法，最后本文给出时间序列决策树 TSDT 与集成分类器 En-TSDT 的构建过程和算法描述。

1.1 问题定义

针对时间序列数据集 D ，样本 $x_i \in D$ 有 $|A|$ 个属性 $a_1, a_2, \dots, a_{|A|}$ ，以及类别属性 $c \in C$ ，其中属性 a_j 代表样本在时刻 j 上的取值， $C = \{c_1, c_2, \dots, c_{|C|}\}$ 表示类别集合。时间序列分类的目的就是根据训练数据集 D ，预测待分类样本 t 的类别 $C(t)$ 。

1.2 序列熵及序列对信息增益

首先给出序列熵与序列对信息增益的定义。序列熵为采用若干时间序列划分时间序列集合时的信息熵。序列对信息增益为采用一个序列对划分时间序列集合时所带来的信息量变化。

由于时间序列数据在时间维度上具有自相关性的特点，不同序列在 j 时刻所代表的信息意义不一定相同，而传统决策树算法则将序列的时刻一一对齐^[12]，导致计算偏差。通过对传统信息熵及信息增益^[13]的扩展，将传统决策树以时间序列的不同时刻作为属性扩展为以不同时间序列作为属性，得到针对时间序列数据的序列熵及序列对信息增益，以便度量属性值为时间序列对的数据所包含的分类信息，从而选择信息增益最大的序列对作为节点的分割属性，解决了时间序列的自相关性和不对齐性对构建决策树时属性选择的影响。在分割序列对对数据集 D 的划分上，将基于动态时间规整算法 (DTW) 的时间序列数据相似度作为距离标准，根据待划分序列与分割对中序列的距离将待划分序列划分至与分割对中距离最近的序列所代表的类别集合中。具体计算方法如下：

在时间序列数据集 D 中，随机选取序列 x_1, x_2 ，其中 $C(x_1)=c_1, C(x_2)=c_2, c_1 \neq c_2$ 。将两个序列整合为序列对 $s=\langle x_1, x_2 \rangle$ ，则该序列对 s 根据数据集 D 中样本与 x_1, x_2 的 DTW 距离，可将 D 中序列划分到 x_1, x_2 所属类别所代表的 $Child_1, Child_2$ 两个子

节点中。序列 x_1, x_2 的序列熵及序列对 s 的信息增益计算如下：

序列对 s 对 D 划分后第 k 个子集的序列熵为

$$H(Child_k) = -\sum_{j=1}^{|C|} P(C(x)=c_j) \cdot \log_2(P(C(x)=c_j)) \quad (1)$$

s 对 D 划分后的序列熵为

$$H(D|s) = \sum_{k=1}^2 P(Child_k) \cdot H(Child_k) \quad (2)$$

该序列对 s 的信息增益为

$$IG(D, s) = H(D) - H(D|s) \quad (3)$$

TSDT 模型

时间序列决策树 (TSDT) 是一种二叉树，每个非叶子节点表示一簇时间序列数据的集合，每个分支代表通过序列对对其父节点的划分，而每个叶子节点存放一个类别。TSDT 进行决策过程就是从根节点开始，根据节点序列对的划分选择分支，直至达到叶子节点，将叶子节点所存放的类别作为决策结果。

TSDT 对包含 4 个类别的时间序列数据集的 TSDT 构建过程为如图 2 所示，先根据序列对 $\langle C1, B1 \rangle$ 将数据集划分为两个子集，针对不同子集再选择序列对信息增益最大的序列对 $\langle A2, B2 \rangle$ 和 $\langle C1, D2 \rangle$ 对两个子集进行划分，直至所划分子集中所有时间序列属于同一个类别。

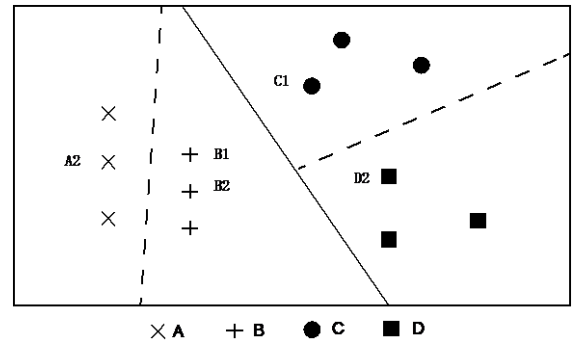


图 2 TSDT 针对多类别数据的构建过程

不同于传统决策树，TSDT 在节点 Node 上的分裂过程如图 3 所示。先从节点 Node 中随机选择两个不同类别的序列 $C(x_1)=c_1, C(x_2)=c_2, c_1 \neq c_2$ ，作为分割序列对 $s=\langle x_1, x_2 \rangle$ 。由于节点中数据可能为多类别，为保证任意两个类别的序列对都可以考察到，在选取时首先按上述方法选取 nbPairs 个候选序列对，然后根据候选序列对的信息增益，选择信息增益最大的序列对 $s=\langle x_1, x_2 \rangle$ 作为分割属性，信息增益越大，所划分子集的纯度越高。然后使用序列对 s 将 Node 划分为两个子集 $\{Child_1, Child_2\}$ ，计算 Node 中每一个序列 x_i 与 x_1, x_2 的 DTW 距离 $d_{x_1} = DTW(x_i, x_1)$ ， $d_{x_2} = DTW(x_i, x_2)$ ，根据两个距离的大小，将序列 x_i 加入到集合 $Child_1$ 或 $Child_2$ 中。

$$\begin{cases} x_i \in Child_1 & d_{x_1} \leq d_{x_2} \\ x_i \in Child_2 & otherwise \end{cases} \quad (4)$$

若 $Child_k$ 为叶子节点，则 $\forall x \in Child_k, C(x)=c_j$ 。

与传统决策树算法相比，TSDT 将一个时间序列数据作为整体，而不是深入到时间序列中的某个时刻。其次，采用序列对将数据集划分成两部分，而不是某一个特定的取值。其属性的

数量与样本数量有关，与序列长度无关。因此，当训练集合较小、时间序列长度较长时，*TSDT* 构建时间小于传统决策树。训练集合较大，而时间序列长度较小时，传统决策树构建时间小于 *TSDT*。算法具体的时空分析将通过实验给出。

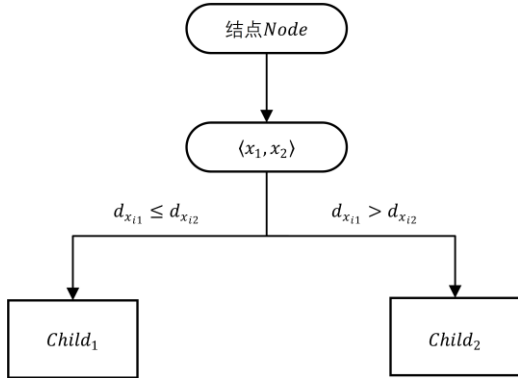


图 3 *TSDT* 在节点 *Node* 的分裂过程

TSDT 具体构建算法如下：

算法 1 时间序列决策树构建算法 *BuildTree(Node, nbPairs)*

输入：数据集 *Node*，候选序列对数量 *nbPairs*

输出：决策树 *TSDT*

- 1 若 $\forall x \in \text{Node}, C(x) = c_j$ ，则该节点为叶子节点
- 2 从 *Node* 中随机选择两个不同类别的序列 x_1 、 x_2 ，组成序列对并存入候选序列对集合中
- 3 重复步骤 2，直到候选集中候选序列对的数量为 *nbPairs*
- 4 根据每个候选数据对的信息增益，选择信息增益最大的候选数据对作为分裂数据对
- 5 使用分裂数据对将 *Node* 划分为 *Child₁* 和 *Child₂* 两个子节点
- 6 对每个子节点 *Child_i*，执行 *BuildTree(Child_i, nbPairs)*

En-TSDT 模型

为得到更好的分类性能，本文采用动态分类器集成技术，构造集成模型 *En-TSDT* [14,15]。首先通过不同初始值构造多个 *TSDT*，然后根据测试样本 *t* 在训练集中寻找到的 *K* 个最近邻，根据基分类器在这 *K* 个邻居上的分类错误率来确定基分类器的权重 [16]。最后根据 *En-TSDT* 对测试样本的加权投票，得到分类结果 [17]。下面给出具体算法。

首先根据算法 1 构建 *n* 个决策树 *TSDT_i*，作为集成分类器的基分类器，然后构建动态集成分类器。构建动态集成分类器主要分为 2 个步骤：1. 动态计算基分类器权重；2. 基分类器加权投票。

计算基分类器权重时，根据输入的时间序列样本 *t*，寻找其在训练集 *D* 中的 *K* 个最近邻的集合 *V*，对每个 *TSDT_i* 进行测试，计算 *TSDT_i* 在最近邻集合 *V* 上的分类错误率 *eR_i*，则 *TSDT_i* 的权重 $w_i = 1 - eR_i$ 。然后计算基分类器的平均错误率 *avg_eR*，如公式(5)所示：

$$\text{avg_eR} = \frac{\sum eR_i}{n} \quad (5)$$

对集成学习算法中基分类器定义如下： $h_{i,j} \in \{0,1\}$ ， $i=1, \dots, n$ ， $j=1, \dots, |C|$ 。 h_i 表示第 *i* 个基分类器，*j* 表示类别。对于时间序列样本 *t*，如果第 *i* 个基分类器将样本分类到第 *j* 个类

别中，有 $h_{i,j}(t)=1$ ，否则有 $h_{i,j}(t)=0$ 。

然后将基分类器中分类错误率满足 $eR_i \leq \text{avg_eR}$ 的分类器加入到集成分类器 *En-TSDT* 中，并将 *En-TSDT* 中的分类器权重规范化，得到 h_i 的权重 w_{i_final} ，见公式(6)。则测试序列 *t* 的分类类别的计算为公式(7)所示。

$$w_{i_final} = \frac{w_i}{\sum w_i} \quad (6)$$

$$C(t) = \text{avg} \max_j \sum_{i=1}^{|En-TSDT|} w_{i_final} h_{i,j}(t) \quad (7)$$

算法 2 动态集成分类器 *En-TSDT(D, t)*

输入：训练集 *D*，测试样本 *t*

输出：分类类别 *c*

- 1 for *i*=1 到 *n*
- 2 对训练集 *D* 进行 50% 随机抽样，得到随机训练集 *sample_D*
- 3 构建基分类器 *TSDT_i* = *BuildTree(sample_D, nbPairs)*
- 4 计算待测样本 *t* 在训练集中 *D* 的 *K* 个最近邻的集合 $V = KNN(t, D)$
- 5 对每个基分类器 *TSDT_i* 测试最近邻集合 *V*，得到每个基分类器的错误率 *eR_i*
- 6 选取在测试集 *V* 上分类错误率 $eR_i \leq \text{avg_eR}$ 的基分类器，作为集成分类器 *En-TSDT*，赋予权重 $w_i = 1 - eR_i$ 并规范化为 w_{i_final}
- 7 使用集成分类器 *En-TSDT* 分类测试样本 *t*，得到多数投票后的分类类别 *c*

2 实验结果与分析

2.1 实验数据集

为验证 *En-TSDT* 的分类性能，本文在 UCR 公共数据集上进行实验 [118]，除去由于计算时间过长而没有进行测试的数据集 *ElectricDevices* 外，共测试了 84 个数据集，由于篇幅有限，这里给出随机选出的 10 个数据集上的测试结果。

2.2 实验结果

为验证 *En-TSDT* 的分类性能，选取 *INN-DTW* 作为对比算法，该算法是时间序列分类领域中的基准分类算法，在时间序列分类上的性能上强于其他传统分类算法 [8]。本文从分类器的准确率和 *F1* 值两个角度进行对比 [19]。在计算 *F1* 值时，将训练集中出现的第一个数据的类别作为目标类，其他类别作为一类。算法均采用 Java 语言编写，在高性能计算平台 (*HPC*) 上测试，且 Java 最大内存设置为 4G。

实验采用 10 次实验取均值的方式，其中集成学习中基分类器的最大数目设为 100，最小数目为 30，候选序列对最大值为 100。参数选择对算法性能的影响将在下一节说明。

在本文随机选择的 10 个数据集上及全部 84 个数据集平均测试结果如表 1 所示。

图 4、5 分别为 84 个数据集在 *INN-DTW* 和 *En-TSDT* 下的错误率和 *F1* 值。

¹ http://www.cs.ucr.edu/~eamonn/time_series_data/

表 1 UCR 数据集错误率与 F1 值

数据集	1NN-DTW		TSDT		En-TSDT	
	错误率	F1 值	错误率	F1 值	错误率	F1 值
50Words	0.30989	0.776119	0.511062	0.559723	0.314286	0.761905
CBF	0.003333	0.996633	0.061259	0.951138	0	1
ECG	0.23	0.821705	0.238667	0.816360	0.12	0.854961
Gun-Point	0.093333	0.905405	0.219555	0.790643	0.073333	0.878378
InlineSkate	0.616364	0.475610	0.709333	0.350399	0.614545	0.397516
InsectWingbeatSound	0.644949	0.597561	0.706279	0.584968	0.462121	0.697297
OliveOil	0.166666	0.727273	0.222222	0.647418	0.1	0.8
Synthetic Control	0.006667	0.979592	0.045778	0.944459	0.016667	0.98990
Wafer	0.020116	0.988819	0.029213	0.983698	0.015574	0.991315
Yoga	0.163333	0.820381	0.259453	0.717742	0.150333	0.833272
84 个数据集平均结果	0.254606	0.745765	0.336963	0.660994	0.229858	0.757005

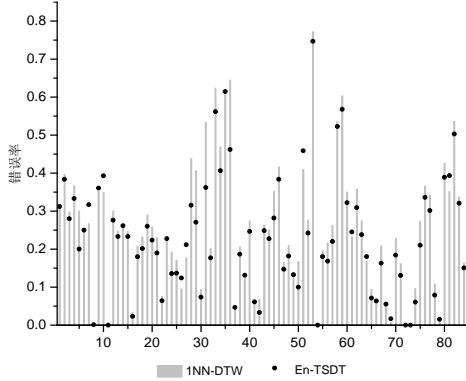


图 4 分类错误率

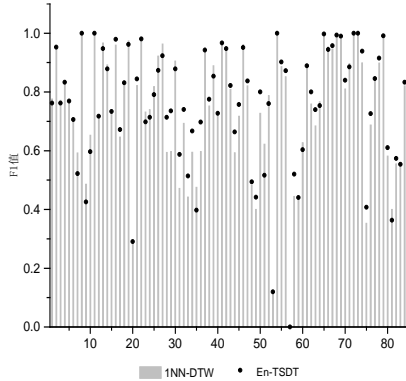


图 5 分类 F1 值

由表 1 中可得出 *En-TSDT* 算法的分类精度绝大部分优于 *1NN-DTW*。*En-TSDT* 在全部数据集上, 共有 83.33% 的数据集测试结果优于或持平于 *1NN-DTW*。从图 4、图 5 可以看出, 在大部分数据集上, *En-TSDT* 的错误率低于 *1NN-DTW*, 而在 *F1* 上两者相差不多, *En-TSDT* 略微胜出。在全部 84 个数据集上, 与 *1NN-DTW* 相比, *En-TSDT* 错误率下降了 9.72%, *F1* 值提高了 1.51%。实验表明 *En-TSDT* 分类性能强于 *1NN-DTW*, 且在不同数据集上均有良好分类性能, 有较强鲁棒性。

2.3 参数分析

基分类器的数量对分类准确率与分类时间均有影响, 本文选取 50words 数据集测试基分类器数量 *n* 与 *En-TSDT* 分类性能和分类时间的关系, 实验结果如图 6、7 所示。

由图 6、7 可以看出, 随着基分类器数量增加, *En-TSDT* 分类错误率越来越低, 而分类所需时间越来越长。这说明分类器集成技术可以有效提高分类性能, 但相比单分类器则需要更多的时间。

候选序列对数量 *nbPairs* 同样会对模型性能产生一定影响, 在 *TSDT* 算法中, 足够多的候选序列对能够保证可以得到任意两个类别的序列对组合中信息增益最大的组合, 但如果不对最大数量进行限制, 则会花费大量的计算时间。因此, 本文中, *nbPairs*

的取值如(8)式所示, 其中 *ratio* 为选取比例, *max* 为最大选取数量。在实验中, 本文设定 *ratio* = 10%, *max* = 100。通过在 50words 数据集上测试, 不同候选序列对选取比例 *ratio* 对分类错误率和分类时间的影响如图 8 和 9 所示。

nbPairs = min(ratio * \prod_{j=1}^{|C|} |{C(x) = c_j}| , max)

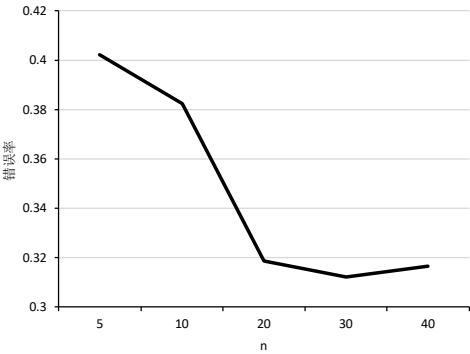


图 6 基分类器数量 n 与 En-TSDT 分类错误率关系

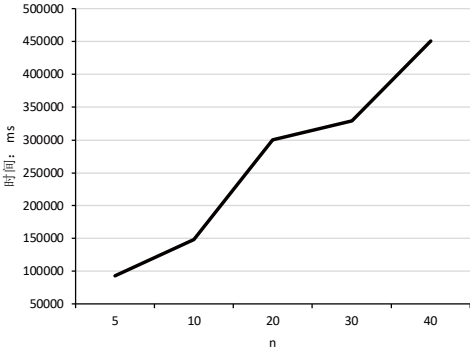


图 7 基分类器数量 n 与 En-TSDT 分类时间关系

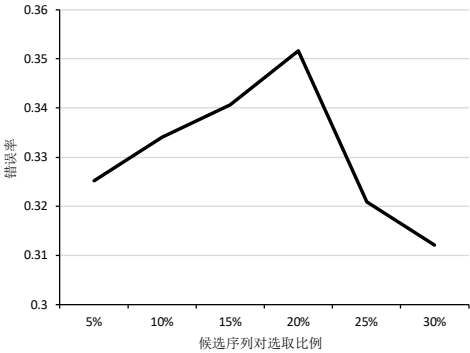


图 8 不同 ratio 下 En-TSDT 分类错误率

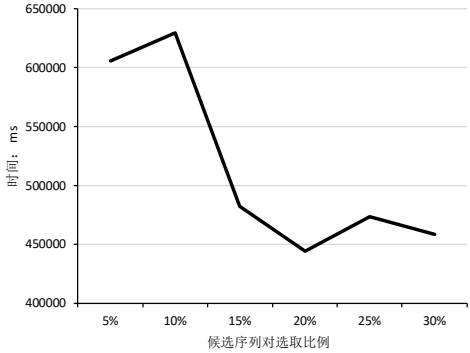


图 9 不同 ratio 下 En-TSDT 分类时间

由图 8 和 9 中可以看出, 不同候选序列对选取比例会对选

取最优分割序列对产生影响, 候选序列对选取比例越大, 分割序列接近最优选择的概率越大。同时, 序列对信息增益越大, 以此序列对分割节点得到的树越简洁, 分类时时间越短。

2.4 时空分析

$1NN-DTW$ 与 $TSDT$ 、 $En-TSDT$ 在分类时间上的比较见表 2。在表 2 中, $TSDT$ 所用时间最少, $En-TSDT$ 用时最多。 $TSDT$ 模型可以有效减少分类时间, 而由于 $En-TSDT$ 模型采用动态分类器集成技术, 需要根据不同的样本自适应调整基分类器权重, 相比 $1NN-DTW$ 而言, 将消耗较多时间在测试基分类器的准确率上, 但是集成分类模型能得到更低的错误率。因此, 通过何种方法缩短分类时间, 将是今后研究的内容之一。

表 2 各模型分类所用时间 (单位: ms)

数据集	1NN-DTW	TSDT	En-TSDT
50Words	85694	2259.4	454884
CBF	3179	1035.5	240571
ECG	516	80.43	14710
Gun-Point	668	131.43	23929
InlineSkate	578631	73169.4	12686559
InsectWingbeatSound	111656	7756.97	1422467
OliveOil	1072	298.07	52978
Synthetic Control	2555	248.7	47379
Wafer	403440	83388.55	52738527
Yoga	570154	36170.93	16436289
84 个数据集平均结果	932788.274	29188.9946	14877069.3

3 结束语

传统决策树算法在时间序列分类上忽视了序列数据自相关性, 基于此, 本文在传统决策树算法中信息熵的基础上, 针对时间序列数据的特点, 提出了序列熵的概念, 根据序列熵计算序列对信息增益, 将信息增益最大的序列对作为分裂属性, 构建适于时间序列分类的决策树模型 $TSDT$, 并以 $TSDT$ 为基分类器, 基于动态集成技术, 提出了集成分类算法 $En-TSDT$ 。在 UCR 数据集上的实验结果表明, 集成分类器 $En-TSDT$ 的分类性能优于目前时间序列数据分类研究领域的基准分类器 $1NN-DTW$, 但是存在分类时间长的问题。如何缩短分类时间将是今后的研究内容之一。

参考文献:

[1] 原继东, 王志海. 时间序列的表示与分类算法综述 [J]. 计算机科学, 2015, 42(3): 1-7.
 [2] Esling P, Agon C. Time-series data mining [J]. ACM Computing Surveys, 2012, 45(1): 1-34.
 [3] Keogh E J, Pazzani M J. Scaling up dynamic time warping for datamining applications[C]//Proc of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2000: 285-289.

[4] 王树英, 王志海. 基于增量式决策树的时间序列分类算法研究[J]. 现代计算机: 专业版, 2015(3): 26-30.
 [5] 刘栋, 宋国杰. 面向多维时间序列的过程决策树模型[J]. 计算机应用, 2011, 31(5): 1374-7.
 [6] Lundkvist E. Decision tree classification and forecasting of pricing time series data [J]. 2014.
 [7] Yamada Y, Suzuki E, Yokoi H, et al. Decision-tree induction from time-series data based on a standard-example split test[C]//Proc of ICML. 2003: 840-847.
 [8] Bagnall A, Lines J. An experimental evaluation of nearest neighbor time series classification, Technical report# cmp-c14-01[J]. 2014.
 [9] Rokach L, Maimon O. Data mining with decision trees: Theory and applications [M]. World scientific, 2014.
 [10] Dietterich T G. Ensemble methods in machine learning [C]//Proc of International Workshop on Multiple Classifier Systems. Berlin: Springer, 2000: 1-15.
 [11] Xi X, Keogh E, Shelton C, et al. Fast time series classification using numerosity reduction[C]//Proc of the 23rd International Conference on Machine Learning. 2006: 1033-1040.
 [12] Quinlan J R. C4. 5: programs for machine learning [M]. San Francisco: Morgan Kaufmann Publishers Inc. , 1993.
 [13] Shannon C E. A mathematical theory of communication [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1): 3-55.
 [14] Britto A S, Sabourin R, Oliverira L E S. Dynamic selection of classifiers—a comprehensive review [J]. Pattern Recognition, 2014, 47(11): 3665-3680.
 [15] Xiao J, Xie L, He C, et al. Dynamic classifier ensemble model for customer classification with imbalanced class distribution [J]. Expert Systems with Applications, 2012, 39(3): 3668-3675.
 [16] Ko A H R, Sabourin R, Jr Britto A S. From dynamic classifier selection to dynamic ensemble selection [J]. Pattern Recognition, 2008, 41(5): 1718-1731.
 [17] Dietterich T G. Ensemble learning [R]//The Handbook of Brain Theory and Neural Networks. 2002, 2: 110-25.
 [18] Chen Y P, Keogh E, Hu B, et al. The UCR time series classification archive[EB/OL]. 2015. http://www.cs.ucr.edu/~eamonn/time_series_data/.
 [19] Powers D M, Ailab. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation [J]. Journal of Machine Learning Technologies, 2008, 2: 2229-398.