第 40 卷 第 4 期 计 算 机 工 程 2014 年 4 月
Vol.40 No.4 Computer Engineering April 2014

·人工智能及识别技术· 文章编号：1000-3428(2014)04-0175-07 文献标识码：A 中图分类号：TP312

一种事件序列的加权变阶马尔可夫模型

吴宏和，陈黎飞，郭躬德

(福建师范大学数学与计算机科学学院，福州 350007)

摘 要：变阶马尔可夫模型是对事件序列建模的一种简单且有效的模型，但经典变阶马尔可夫模型只考虑转移概率，未关注子序列本身出现的频率。为此，提出一种加权的变阶马尔可夫模型，在经典变阶马尔可夫模型基础上根据子序列的频率构建一棵加权概率后缀树。给出一种剪枝策略，在构建后缀树时根据结点相似程度剪除树枝，以提高模型的泛化能力，并在线性时间内完成加权概率后缀树的构建。通过将加权的模型应用于事件序列分类进行实验验证，结果表明，该模型可以对不同领域的实际序列数据进行有效分类。

关键词：变阶马尔可夫模型；概率后缀树；事件序列；分类；加权；剪枝

A Weighted Variable Order Markov Model for Event Sequences

WU Hong-he, CHEN Li-fei, GUO Gong-de

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

【Abstract】Variable-order Markov Model(VLMM) is a simple but effective model for event sequences modeling. However, the classic VLMM only considers the transition probability, without taking into account the frequency of the substring. A Weighted VLMM(WVLMM) is proposed in this paper, constructing a Weighted Probabilistic Suffix Tree(WPST) via the frequency of the substring based on the classic VLMM. It also proposes a strategy for branches pruning based on the degree of the similarity of the nodes while constructing the tree, in order to improve the generalization ability of the model, and to construct the tree in a linear time complexity. To validate the effectiveness of the model, the proposed model is applied to the classification of event sequences. Experimental results demonstrate that the new model can make an effective classification on real-world sequence datasets in different applications.

【Key words】Variable-order Markov Model(VLMM); probabilistic suffix tree; event sequence; classification; weighted; pruning

DOI: 10.3969/j.issn.1000-3428.2014.04.034

1 概述

数据挖掘作为一个新兴的研究领域，自 20 世纪 80 年代开始以来，已经取得了显著进展并且涵盖了广泛的应用。如今，数据挖掘已经被应用到了众多的领域，同时出现了大量商品化的数据挖掘系统和服务。然而，许多挑战依然存在，如复杂数据类型的数据挖掘，序列数据的分类就是一项具有挑战性的任务[1]。

序列是事件的有序列表，事件序列是数据的一种常见形式，广泛存在于数据挖掘的各个领域，如蛋白质序列数据、顾客在商场的购买活动记录、用户访问网站的点击流等[2-3]。如何对这些事件序列正确分类，是数据挖掘研究的一项重要内容。在事件序列多种类型的分类方法中，基于模型的一类方法[4]由于能够有效挖掘序列中潜在统计特性的优点而备受关注。这种方法是建立在使序列模型化基础之上，已有隐马尔可夫模型[5]、变阶马尔可夫模型(Variable-order Markov Model, VLMM)[6]等统计学模型被应用于事件序列建模，其中，变阶马尔可夫模型具有模型简单、构建的时间复杂度低且有效表达序列特征等优点[7]，自提出以来得到了广泛的研究和应用。

文献[8]提出变阶马尔可夫模型，文献[6]将此模型引进到序列数据的建模领域。随后，有不同方面的改进，文献[9]从理论上证明了变阶马尔可夫模型能够在线性时间与空间内构建的可行性，文献[7]从算法上具体实现了模型的线性时间构建。文献[10]实现了模型的在线学习算法。文献[4]使用数组代替后缀树来构建模型降低了模型的时间与空间消耗。文献[11]针对序列中某些元素的稀疏性提出稀疏马尔可夫模型等。但它们都未把序列中子序列的频率这个统计特性增加到模型中，且缺乏对概率后缀树有效的剪枝策略，从而影响了模型的泛化能力。

针对上述问题，本文提出一种新的变阶马尔可夫模型——加权变阶马尔可夫模型(Weighted Variable-order

基金项目：国家自然科学基金资助项目(61175123)。

作者简介：吴宏和(1987－)，男，硕士研究生，主研方向：数据挖掘；陈黎飞，副教授、博士；郭躬德，教授、博士。

收稿日期：2013-07-11 **修回日期：**2013-08-14 **E-mail：**leopardsaga@gmail.com

Markov Model, WVLMM)，用于事件序列建模，并进行序列分类。新模型在经典变阶马尔可夫模型的基础上增加了子序列的权重，以增强模型对序列特征的表达。同时，在构建模型的加权概率后缀树(Weighted Probabilistic Suffix Tree, WPST)时进行有针对性的提前剪枝，以提高模型的泛化能力。

2 背景知识与相关工作

本节讨论事件序列建模相关的背景知识并介绍与分析若干相关工作。首先约定全文使用的记号。

定义 1 设 σ_i 表示一个事件，记 N 个事件的集合为 $\Sigma=\{\sigma_1,\sigma_2,\cdots,\sigma_N\}$。

定义 2 事件序列 S 是由集合 Σ 中的成员组成的有序序列 $S=s_1s_2\cdots s_m$，其中，每个 s_i 是 Σ 的一个成员，称为 S 的一个元素； m 称为 S 的长度，表示为 $|S|$； $1\sim m$ 之间的数字表示 S 中的位置。

序列建模方面已提出多种模型，如变阶马尔可夫模型[12]、隐马尔可夫模型[13]、支持向量机[14]以及神经网络[15]等。在基于模型的序列分类方法中，文献[8]提出了变阶马尔可夫模型作为数据压缩的通用模型，同时，也为这类模型引进了估计算法——CA(子序列算法)。文献[16]证明了这个算法的一致性。这意味着，假设数据由变阶马尔可夫模型生成，在足够大的样本下，CA 能够找到有效的模型来表达序列。

概率后缀树[6]是另一个实现变阶马尔可夫模型的算法，并且比 CA 算法计算复杂度低[9]，已成功应用在蛋白质序列分类领域[12]。针对概率后缀树算法实现的时间与空间复杂度高的问题，文献[9]从理论上证明了在线性时间与空间复杂度内构建概率后缀树的可行性，而文献[7]使用懒后缀树与增强后缀数组，使概率后缀树在线性时间与空间复杂度内成功实现构建。随后，文献[4]提出变阶马尔可夫模型的另一种数据结构实现——概率后缀数组(Probabilistic Suffix Array, PSA)，解决了时间与空间部分问题，但不能在线性时间内将模型应用于序列分类。

针对某些序列元素的稀疏性(如蛋白质序列)，文献[11]提出了变阶马尔可夫模型的变体——稀疏马尔可夫模型(SMC)。稀疏马尔可夫模型与变阶马尔可夫模型的区别在于，模型中的条件转移概率是基于序列事件集合的子集，而变阶马尔可夫模型是基于序列事件集合的全集；同时，文献[11]引进了概率后缀树算法的变体稀疏概率后缀树来估计序列的稀疏马尔可夫模型，然而，这种模型与算法实现时间复杂度高。针对原始概率后缀树算法需人为预先设定变阶马尔可夫模型阶长的不足，文献[10,17]分别在概率后缀树构建时引进 winnow 与 perceptron 算法，使概率后缀树能自适应估计模型的阶来构建变阶马尔可夫模型，实现了模型构建算法的在线学习。然而这些改进的算法，皆只在单一类型数据集上测试，未充分考虑模型泛化能力。

本文针对经典概率后缀树算法构建出的变阶马尔可夫模型泛化能力较弱的问题提出改进：

(1)提出加权变阶马尔可夫模型 WVLMM，为原始马尔可夫模型增加子序列权重因子，增强模型对序列特征的表达；

(2)对模型的加权概率后缀树 WPST 实现，在其构建时提前剪枝，增强模型的泛化能力。

3 模型及算法描述

本节详细阐述提出的加权变阶马尔可夫模型 WVLMM 及模型的实现算法——加权概率后缀树 WPST。首先简要描述经典变阶马尔可夫模型及其在序列上的表示；然后具体叙述 WVLMM 模型与 WPST 及其剪枝算法。最后分析了 WPST 算法的时间复杂度，指出新算法可以在相对于序列长度的线性时间内构建出加权概率后缀树。

3.1 变阶马尔可夫模型

变阶马尔可夫模型[11]是一个平稳随机过程 $\{X_n,n=0,1,\cdots,X_i\in A\}$，其中， A 为有限状态集 $\{x_i\}$，满足：

$$P[X_n=x_n\mid X_0=x_0,X_1=x_1,\cdots,X_{n-1}=x_{n-1}]=$$
$$P[X_n=x_n\mid X_{n-l}=x_{n-l},\cdots,X_{n-1}=x_{n-l}]$$

其中，l 是模型的阶，有限序列 $(x_{n-l},\cdots,x_{n-1})$ 称为子序列。使序列的事件集合 Σ 对应于状态集 A，则序列可以看成一个平稳随机过程，相应的用变阶马尔可夫模型可表示如下：

$$P[s_n=\sigma_n\mid s_0=\sigma_0,s_1=\sigma_1,\cdots,s_{n-1}=\sigma_{n-1}]=$$
$$P[s_n=\sigma_n\mid s_{n-l}=\sigma_{n-l},\cdots,s_{n-1}=\sigma_{n-l}]$$

变阶马尔可夫模型拥有后缀特性，即模型中不可能出现一个子序列是另一个子序列的后缀。因此，元素转移概率的分布不会出现模糊混淆的情况。

同时，后缀特性使得可以使用树这种数据结构来表示模型，图 1 以序列 10001 00110 为例，给出了该序列基于变阶马尔可夫模型构建的概率后缀树表示。其中，树的左右 2 个边分别表示元素 0 与 1，结点表示从此结点至根结点路径上元素组成的序列，结点旁边的向量为与其相关的转移概率分布向量，如结点 00 的转移概率分布向量(1/3, 2/3)表示 $P(0|00)=1/3$, $P(1|00)=2/3$。

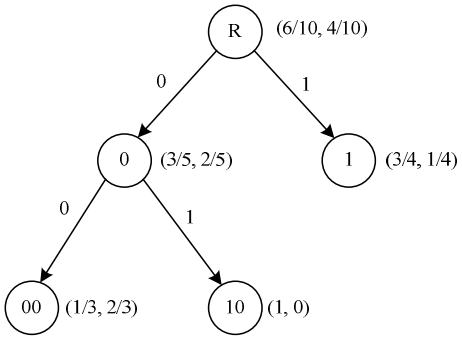

图 1 由序列 1000100110 生成的变阶马尔可夫模型

3.2　加权变阶马尔可夫模型

从 3.1 节的描述可以看出，经典变阶马尔可夫模型只考虑转移概率，而没有考虑子序列本身出现的频率。对于序列这种复杂的数据类型，如果同时考虑元素的转移概率与子序列的频率，则能更准确地用马尔可夫模型描述序列。以序列 A:100100101101 与序列 B:100101111111 为例，虽然从 10 转移到 1 的概率，序列 A 为 $P(1|10)_A = 2/4 = 1/2$，序列 B 为 $P(1|10)_B = 1/2$，具有相同的数值 1/2。但子序列 10 的频率，序列 A 为 $P(10)_A = 4/11$，序列 B 为 $P(10)_B = 2/11$，序列 A 中的子序列 10 的频率明显比序列 B 高，如果对 2 个序列构建的变阶马尔可夫模型增加子序列 10 的相应权重(一个与其频率相关的因子)，则模型就能够表示更为准确的序列信息。因此，本文提出变阶马尔可夫模型的改进——加权变阶马尔可夫模型，对模型增加子序列的频率因子。序列的加权变阶马尔可夫模型如下：

$$P[s_n = \sigma_n \mid s_0 = \sigma_0, s_1 = \sigma_1, \cdots, s_{n-1} = \sigma_{n-1}] = P[s_n = \sigma_n \mid s_{n-l} = \sigma_{n-l}, \cdots, s_{n-1} = \sigma_{n-l}] \times \omega$$

其中，ω 是与子序列频率相关的因子。图 2 显示由 1000100110 构造的加权变阶马尔可夫模型，图中结点的 2 个标注 3/9 分别表示对应 2 个结点(模型的子序列)的权重。图 2 与图 1 对比可以看出，加权变阶马尔可夫模型除了拥有序列元素的转移概率，另增加了 2 个高频子序列 00 与 10(即加权概率后缀树的 2 个结点)的频率。

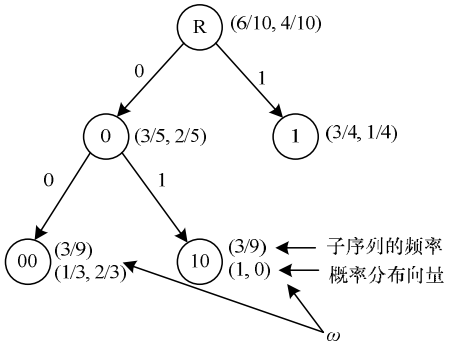

图 2　由序列 1000100110 生成的加权变阶马尔可夫模型

3.3　加权概率后缀树的剪枝算法

根据 3.2 节提出的加权变阶马尔可夫模型，本节描述由加权概率后缀树实现加权变阶马尔可夫模型的构建，并提出一种在构建过程中有针对性提前剪枝的策略。

加权概率后缀树是一棵非空树，结点的度取值空间为 $\{n \mid 0 \leqslant n \leqslant |\Sigma|\}$。每个边由集合 Σ 中的一个成员标注，每个结点由从结此点到根结点路径上的元素组成的序列标注。每个结点都有一个 Σ 上的转移概率分布向量与结点所代表子序列的频率因子。当概率后缀树对未知序列计算相似度时，需要使用结点的这些信息。结点的转移概率分布向量为集合 Σ 各元素相对于结点所代表子序列的转移概率，结点的频率为子序列在序列中出现的经验概率。一棵加权概率后缀树的生成如图 3 所示。

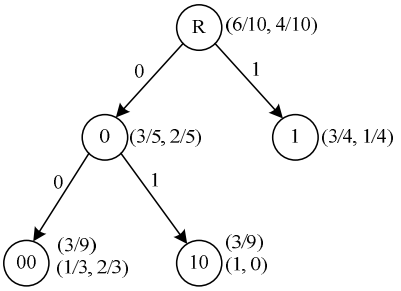

图 3　由序列 1000100110 生成的加权概率后缀树

在描述加权概率后缀树的剪枝算法前，约定一些记号。设序列事件集合 $\Sigma = \{0,1\}$，$\{r^1, r^2, \cdots, r^m\}$ 为在 Σ 上构建的 m 个序列的样本集，其中第 $i(i=1,2,\cdots,m)$个序列的的长度为 l_i。在给定样本集上的子序列的经验概率为 $\tilde{P}(s)$，表示在样本集中观察到子序列的次数除以一个同样长度模式可能出现的最大次数。形式上表示，给定一个长度为 l 的子序列 $s = s_1 s_2 \cdots s_l$，定义一个集合含有元素：

$$\chi_s^{i,j} = \begin{cases} 1 & \text{if } s_1 s_2 \cdots s_l = r_j^i r_{j+1}^i \cdots r_{j+(l-1)}^i \\ 0 & \text{otherwise} \end{cases}$$

对于每个 $i = 1,2,\cdots,m$ 与 $j = 1,2,\cdots,l_i - (l-1)$，当且仅当 s 是 r^i 从 j 位置开始的子序列，指标变量值为 1，图 4 为 s 与 r^i 在位置 j 的匹配，此时指标向量 $\chi_s^{i,j} = 1$。

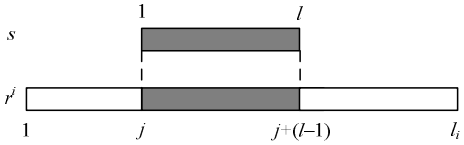

图 4　s 与 r^i 在位置 j 的匹配

序列集 $\{r^i\}$ 中子序列 s 出现的次数为 $\chi_s = \sum_{i,j} \chi_s^{i,j}$，长度为 $|s| = l$ 的子序列可能出现的最大次数为：

$$N_s = \sum_{i \text{ s.t. } l_i \geqslant l} (l_i - (l-1))$$

子序列 s 的经验概率为 $\tilde{P}(s) = \chi_s / N_s$。继续约定一个元素紧跟在一个给定的子序列后面的转移概率 $\tilde{P}(\sigma \mid s)$，其为这个元素紧跟在给定子序列后面的次数除以给定子序列出现的总次数，即 $\tilde{P}(\sigma \mid s) = \chi_{s\sigma} / \chi_s$。

约定构建加权概率后缀树过程中用到的一些符号：L 为阶的大小，$P_{\min}$ 为序列出现的最小概率阈值，r 为候选结点与其父结点关于下一个元素转移概率的差异度量，$\gamma_{\min}$ 为平滑因子，$\overline{T}$ 表示加权概率后缀树，$\overline{S}$ 表示待检查的序列集，$\overline{\gamma}_s$ 表示与结点 s 关联的转移概率分布向量，ω 为结点 s 的权重，$suf(s) = s_2 s_3 \cdots s_l$。

WPST 的构建是从单个元素的子序列开始，逐步遍历所有长度从 1 到 L 的可能的子序列。当子序列的经验概率低于一个确定的阈值 $P_{\min}$，或者达到最大长度 L 时，就放弃断续延伸这个子序列，$P_{\min}$ 阈值避免了指数级搜索空间。在构建的开始阶段，初始化一棵只包含一个根结点的加权

概率后缀树。然后对每一个待检查的子序列 s，检查是否有元素在 Σ 中，这个元素对于 s 的转移概率大于 $\gamma_{\min}$，且此概率与这个元素对于 s 的后缀序列 $suf(s)$ 的转移概率的差异大于 r。当这 2 个条件满足时，添加此子序列到树 $\overline{T}$ 中。经过以上构建时剪枝，WPST 构建完成。图 5 是剪枝效果的一个例子。

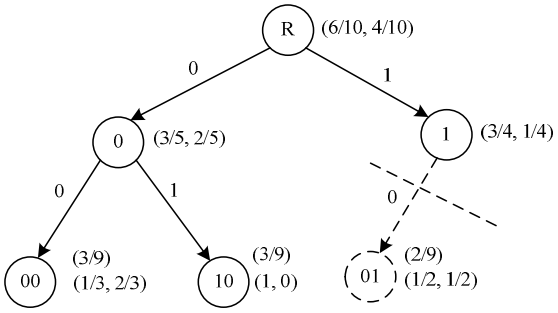

图 5 剪枝后的加权概率后缀树

简单地平滑这些概率，使 Σ 中的元素对于任意子序列的转移概率都不为 0，以便对未知序列计算相似度。综上，构建加权概率后缀树 WPST 算法的伪代码如下：

算法 Build-WPST $(P_m, \gamma_{\min}, r, L, \omega)$

输入 训练集 $\{r^1, r^2, \cdots, r^m\}$，事件集合 Σ

输出 WPST 树

(1)初始化只包含根结点的 WPST 树 $\overline{T}$，$\overline{S} \leftarrow \{\sigma \mid \sigma \in \Sigma, \tilde{P}(\sigma) \geqslant P_{\min}\}$。

(2)当 $\overline{S} \neq \varnothing$，取 $s \in \overline{S}$ 并从 $\overline{S}$ 中删除 s：

1)若 $\exists\, \sigma \in \Sigma$，满足：

① $P(\sigma \mid s) \geqslant \gamma_{\min}$

② $\tilde{P}(\sigma \mid s) / \tilde{P}(\sigma \mid suf(s)) \geqslant r$ or

$\tilde{P}(\sigma \mid s) / \tilde{P}(\sigma \mid suf(s)) \leqslant 1/r$

则添加序列 s 至树 $\overline{T}$，同时添加 ω 至 s 所代表的结点上；

2)若步骤 1)成立且 $|s| < L$，$\forall \sigma' \in \Sigma$，满足 $\tilde{P}(\sigma' s) \geqslant P_{\min}$，则添加 $\sigma' s$ 至 $\overline{S}$。

(3)平滑预测概率：对任意一个表示树 $\overline{T}$ 中结点的序列 s，使：

$$\overline{\gamma}_s(\sigma) = (1 - |\Sigma| \gamma_{\min}) \tilde{P}(\sigma \mid s)(1 + \omega) + \gamma_{\min}$$

3.4 加权概率后缀树快速算法的构建

加权变阶马尔可夫模型的 WPST 实现能在 $O(|S|)$ ($|S|$ 为序列的长度)的时间复杂度下构建[7-9]，但 WPST 是基于后缀树(ST)的数据结构构建，其与概率后缀树(PST)表达形式不一样，先阐述 ST 与 PST 结构上的区别。

PST 树与 ST 树都以边表示元素，以结点表示元素组成的子序列。如图 6 所示，2 个子图表示同一个模型，图 6(a)为 PST 树，图 6(b)为 ST 树。树的结点生出 2 个边，左边的边代表元素 0，右边的边代表元素 1。PST 树结点代表的序列是从此结点开始往上溯回到根结点的路径上所有边所代表元素组成的序列，而 ST 树结点代表的序列是从根结点遍历到此结点路径上所有边代表的元素组成的序列。

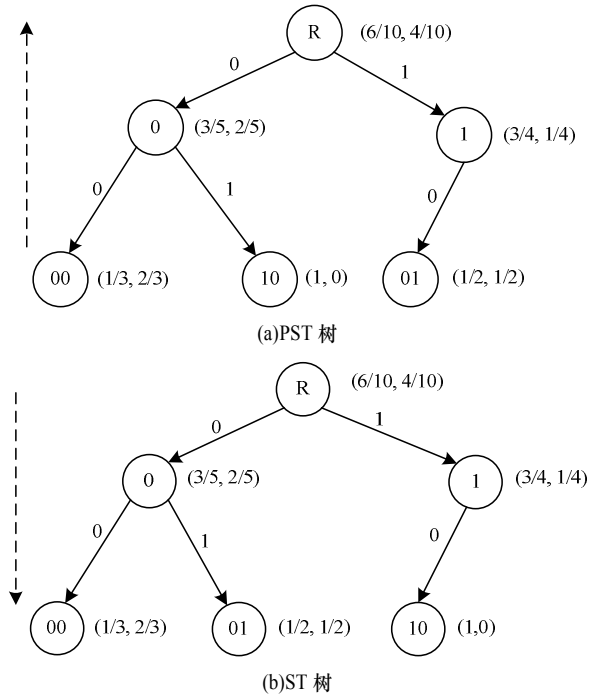

图 6 由序列 1000100110 生成的 PST 树与 ST 树

在描述 WPST 快速构建算法之前，将算法 Build-WPST $(P_m, \gamma_{\min}, r, L, \omega)$ 的剪枝过程分解为 2 个部分：支持度减枝与相似度减枝。

(1)支持度减枝：使 $\overline{T}$ 只包含根结点，$\overline{S} \leftarrow \{\sigma \mid \sigma \in \Sigma, \tilde{P}(\sigma) \geqslant P_{\min}\}$。当 $\overline{S} \neq \varnothing$ 时，执行：

1)从 $\overline{S}$ 中删除 s；

2)将序列 s 添加至树 $\overline{T}$ 中；

3)若 $|s| < L$，$\forall \sigma' \in \Sigma$，满足 $\tilde{P}(\sigma' s) \geqslant P_{\min}$，则将 $\sigma' s$ 加至 $\overline{S}$ 中。

(2)相似度减枝：从根结点往下递归，若不满足以下条件，则剪除 s 代表的结点及其子结点：

1) $P(\sigma \mid s) \geqslant \gamma_{\min}$；

2) $\tilde{P}(\sigma \mid s) / \tilde{P}(\sigma \mid suf(s)) \geqslant r$ or

$\tilde{P}(\sigma \mid s) / \tilde{P}(\sigma \mid suf(s)) \leqslant 1/r$。

图 7 展示了线性时间内由序列 1000100110 构建 WPST 的快速算法。图 7(a)在线性时间内生成 ST[18]并进行支持度减枝。图 7(b)从根结点遍历生成的 ST 树，并计算出各结点所代表子序列的频率与其转移概率分布向量，此亦可在线性时间内完成。图 7(c)在线性时间内增加反向后缀链接[19]，使遍历时能够从结点 r 直接跳转到结点 σr，使构建的 WPST 对未知序列计算相似性时能够在线性时间内完成。图 7(d)进行相似度减枝。至此，以 ST 树加反向后缀链接的 WPST 构建完成，总的模型构建时间复杂度为 $O(|S|)$ ($|S|$ 为训练序列的长度)。从而，WPST 对长度为 m 的新序列的相似性计算时间复杂度为 $O(m)$，而 PST 为 $O(mL)$ 时间(L 为 VLMM 模型中阶的长度)。

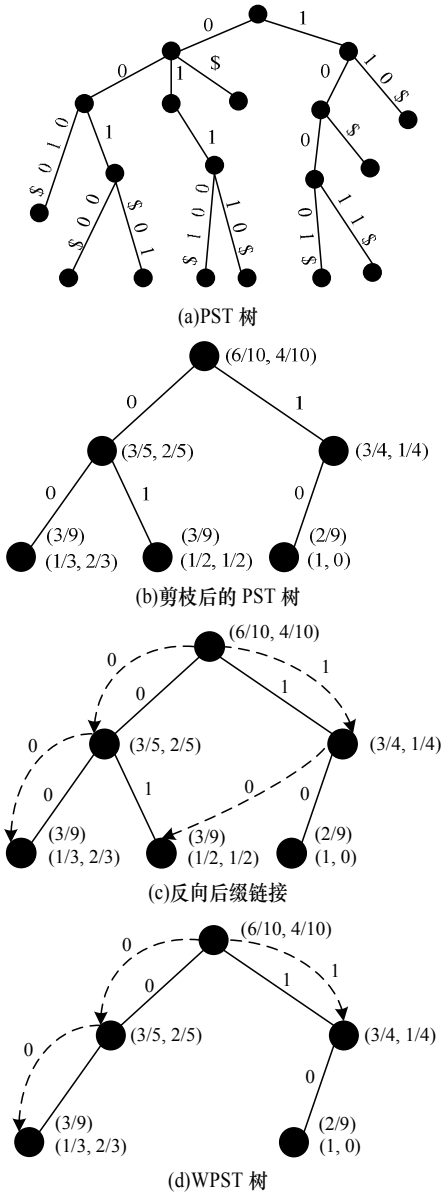

图 7　WPST 树线性时间构建与剪枝过程

4　实验与结果分析

本节通过实验来验证提出的加权变阶马尔可夫模型 WVLMM 及其实现算法加权概率后缀树 WPST 的有效性。选择变阶马尔可夫模型的实现算法概率后缀树(PST)[12]，稀疏马尔可夫模型的实现算法稀疏概率后缀树(SPST)[11]这 2 个马尔可夫模型为对比对象，并通过它们在序列分类中的应用来检验模型的有效性，以 Macro-F1 指标[20-21]作为评价标准。实验为每一类别的训练数据构建一棵概率后缀树，然后以 1NN[22]进行分类，通过比较概率后缀树的分类效果来评估模型对序列特征的表达能力。实验设备配置为：Dell Latitude E6400，CPU Intel P8700 2.5 GHz，RAM 4 GB，Windows 7 Ultimate 32 Byte。

4.1　实验数据

实验采用 4 个数据集，表 1 统计了各数据集的参数，图 8 展示各数据集组成元素的分布比例。

表 1　实验数据集参数汇总

数据集	类数目	序列数目	事件数目	平均长度	最大长度	最小长度
Pfam25.0	20	6 000	20	147	470	18
Voice	5	500	20	1 101	3 753	179
Smart House	6	60	35	231	470	46
DNA	6	119	4	709	3 351	10

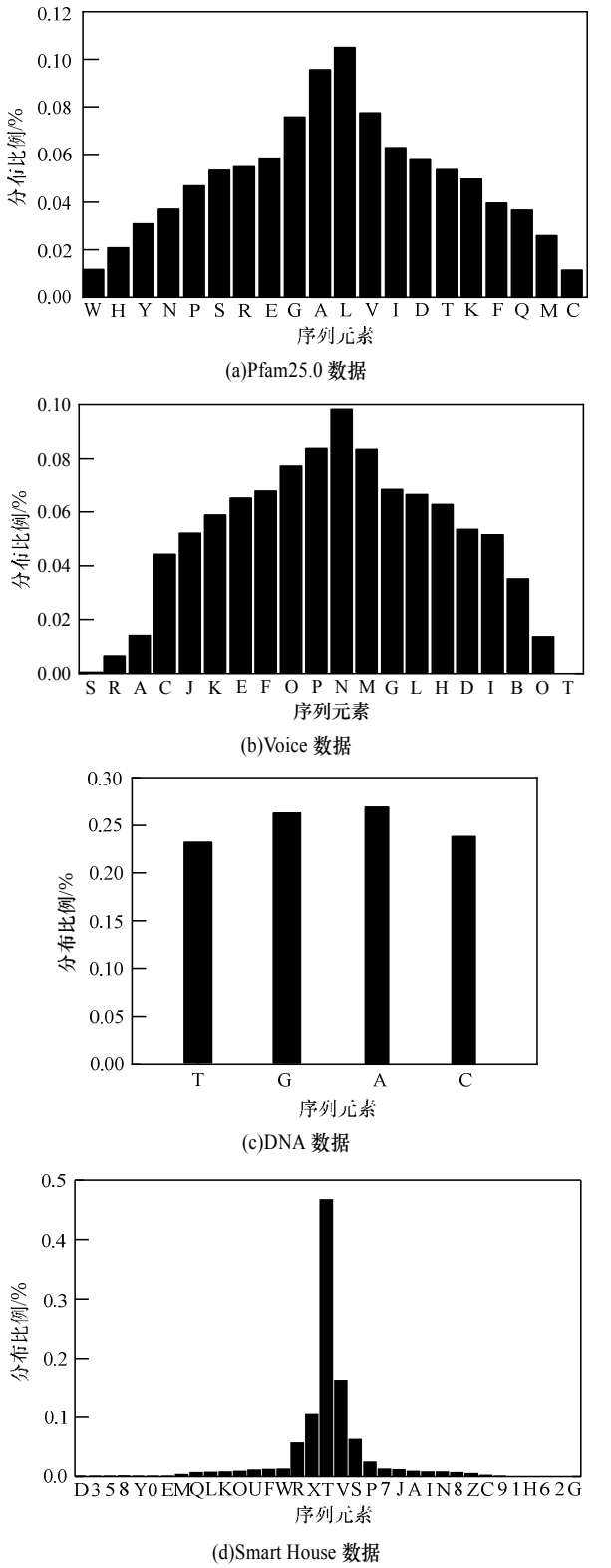

图 8　数据集中序列各元素的分布情况

Pfam25.0[23]数据集为 PST[12]与 SPST[11]算法皆使用的蛋白质序列数据集，并按字母表顺序取 Pfam25.0 前 20 个蛋白质家族，即此数据集取 20 个类别，每个类别取 300 个序列。Voice[24-25]数据集为室内语音数据，共采集 10 个测试人员的发音，数据中的序列从 5 个独立法语元音(‘a’,‘e’,‘i’,‘o’,‘u’)的发音转换而来，每个元音发音 10 次，因此，每个测试人员的发音共有 50 个序列。Smart House[26]为智能家居数据集，来自同一栋房子里 6 个居住者的日常早晨活动，使用住房内 32 个不同传感器记录居住者的活动特征，实验共采集 10 天的数据。DNA 数据集[27]为从 PBIL 的一个同源脊椎动物基因库 HOVERGEN 中抽取出的 6 个类别序列，共有 119 个序列。

4.2 实验结果

实验采用 5-折验证法。通过随机抽样将每个数据集均分为 5 个子集，每次选择其中的 4 个子集为训练数据，剩余的第 5 个子集为测试数据。实验中各算法参数设置为：PST 算法取文献[12]中的参数($P_{\min}=0.000\ 1$, $\alpha=0$, $\gamma_{\min}=0.001$, $r=1.050$, $L=20$)，SPST 取文献[11]中的参数 ($L=20, N_{\min}=3, \gamma_{\min}=0.001, r_{\max}=3.8$) ，WPST 取参数 ($P_{\min}=0.000\ 1, \gamma_{\min}=0.001, r=1.050, L=20, \omega=\log(\tilde{P}(s)/P_{\min})$) 。实验结果中 3 个模型分类效果以 Macro-F1[20-21]指标对比，具体数据如表 2 所示。

表 2 3 种算法分类效果的 Macro-F1 指标对比

算法	Pfam25.0	Voice	Smart House	DNA
PST	0.929	0.966	0.762	0.840
SPST	0.934	0.988	0.768	1.000
WPST	0.954	0.982	0.807	0.909

如表 2 所示，SPST 与 WPST 对序列的预测效果皆优于原始 PST 算法，且 SPST 与 WPST 总的预测能力相当。因 PST 算法构建的模型过拟合现象较明显，而 SPST 与 WPST 对模型的改进增强了其泛化能力，SPST 的泛化方法为对预测能力相近的子序列进行合并，WPST 的泛化技术为对出现频率低于一定阈值的子序列直接剪枝，因此，后两者算法对其他序列数据集拥有更佳的建模与分类效果。同时，表 2 表明 WPST 算法在 Pfam25.0 与 Smart House 数据集上的分类效果明显优于 SPST。从表 1 中的数据参数信息中可以看出，Pfam25.0 与 Smart House 数据集序列长度相对较短、序列中事件数目较多，且图 8 信息表明，两者数据集中少数几个元素占有大部分比率，即序列信息集中于少数元素上。由于 WPST 算法对构建的模型中高频子序列增加了权重，因此在序列较短且序列数据不多的情况下(如 Smart House 数据集)，模型也能有效地表达数据的特征，即 WPST 在序列数据集较小的情况下也能有效地构建模型以表达序列特征。而在 DNA 数据集中 SPST 分类效果很好，如表 1 所示 DNA 序列的事件只有 4 种，因而 SPST 构建的 PST 树每个结点分枝较少，合并分枝的效果较理想，即 SPST 对序列数据中类别的个数与事件数目个数较少的情况下分类效果较好。

同时，为了检验 WPST 与 SPST 算法构建模型的时间复杂度，在一组不同长度的序列上分别统计 2 个算法的使用时间。图 9 为 WPST 与 SPST 两者构建模型的时间对比，构建数据均为蛋白质数据集 2-Hacid_dh 家族序列，取一组不同长度的 6 个序列，分别统计 WPST 与 SPST 算法构建过程的时间消耗。从图 9 可以看出，WPST 算法能在 $O(|S|)$ ($|S|$ 为序列长度)的线性时间内实现模型的构建。而 SPST 因其构建时需进行复杂的结点相似性对比，目前只能在 $O(|S|^2 L)$ (L 为变阶马尔可夫模型阶的长度)内完成构建。综上，加权概率后缀树算法 WPST 在时间要求较高的多事件数目序列的建模与分类方面有较大的优势。

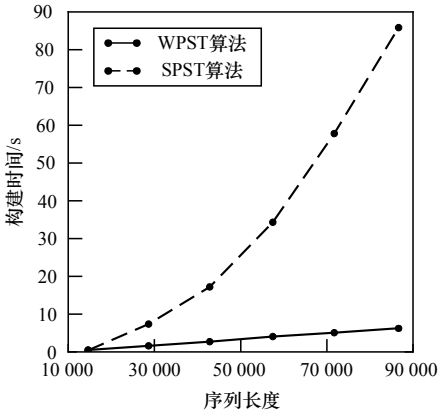

图 9 WPST 与 SPST 算法在不同长度序列上的构建时间对比

5 结束语

本文在经典变阶马尔可夫模型基础上增加子序列的频率因子，提出了加权变阶马尔可夫模型，该模型能更准确地表达复杂事件序列的特性。对加权变阶马尔可夫模型的实现——加权概率后缀树进行构建时提前进行相似性剪枝，提高了模型的泛化能力，并在相对于序列长度的线性时间内完成模型的构建。在蛋白质序列集 Pfam25.0 等公开数据集上的实验结果表明，改进的模型增强了序列特性的表达能力，包含更丰富的统计特性，能够有效地进行分类。下一步工作是深入分析各参数对模型构建的影响，研究数据流的在线学习算法。

参考文献

[1] Han Jiawei, Kamber M. 数据挖掘: 概念与技术[M]. 范 明, 孟小峰, 译. 北京: 机械工业出版社, 2012.

[2] 王清毅, 刘 洁, 蔡庆生. 事件序列中的知识发现研究[J]. 小型微型计算机系统, 1999, 20(1): 16-19.

[3] 刘培华, 王立宏. 一种改进的事件序列相似性计算公式[J]. 计算机工程与应用, 2009, 45(7): 68-69.

[4] Lin Jie, Adjeroh D, Jiang Binghua. Probabilistic Suffix Array: Efficient Modeling and Prediction of Protein Families[J]. Bioinformatics, 2012, 28(10): 1314-1323.

[5] Eddy S R. Hidden Markov Models[J]. Current Opinion in Structural Biology, 1996, 6(3): 361-365.

[6] Ron D, Singer Y, Tishby N. The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length[J]. Machine Learning, 1996, 25(2): 117-149.

[7] Schulz M H, Weese D, Rausch T, et al. Fast and Adaptive Variable Order Markov Chain Construction[C]//Proc. of the 8th International Workshop on Algorithms in Bioinformatics. Karlsruhe, Germany: [s. n.], 2008: 306-317.

[8] Rissanen J. A Universal Data Compression System[J]. IEEE Transactions on Information Theory, 1983, 29(5): 656-664.

[9] Apostolico A, Bejerano G. Optimal Amnesic Probabilistic Automata or How to Learn and Classify Proteins in Linear Time and Space[J]. Journal of Computational Biology, 2000, 7(3/4): 381-393.

[10] Karampatziakis N, Kozen D. Learning Prediction Suffix Trees with Winnow[C]//Proc. of the 26th Annual International Conference on Machine Learning. [S. l.]: ACM Press, 2009: 489-496.

[11] Leonardi F G. A Generalization of the PST Algorithm: Modeling the Sparse Nature of Protein Sequences[J]. Bioinformatics, 2006, 22(11): 1302-1307.

[12] Bejerano G, Yona G. Variations on Probabilistic Suffix Trees: Statistical Modeling and Prediction of Protein Families[J]. Bioinformatics, 2001, 17(1): 23-43.

[13] Ephraim Y, Dembo A, Rabiner L R. A Minimum Discrimination Information Approach for Hidden Markov Modeling[J]. IEEE Transactions on Information Theory, 1989, 35(5): 1001-1013.

[14] Steinwart I, Christmann A. Support Vector Machines[M]. New York, USA: Springer-Verlag, 2008.

[15] Hassoun M H. Fundamentals of Artificial Neural Networks[M]. Cambridge, USA: MIT Press, 1995.

[16] Bühlmann P, Wyner A J. Variable Length Markov Chains[J]. The Annals of Statistics, 1999, 27(2): 480-513.

[17] Dekel O, Shalev S S, Singer Y. The Power of Selective Memory: Self-bounded Learning of Prediction Suffix Trees[C]//Proc. of Conference on Neural Information Processing Systems. [S. l.]: Springer, 2004: 345-352.

[18] Ukkonen E. On-line Construction of Suffix Trees[J]. Algorithmica, 1995, 14(3): 249-260.

[19] Maaß M G. Computing Suffix Links for Suffix Trees and Arrays[J]. Information Processing Letters, 2007, 101(6): 250- 254.

[20] van Rijsbergen C J. Information Retrieval[M]. [S. l.]: Butterworths, 1979.

[21] 陈黎飞，郭躬德. 最近邻分类的多代表点学习算法[J]. 模式识别与人工智能, 2011, 24(6): 882-888.

[22] Cover T, Hart P. Nearest Neighbor Pattern Classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27.

[23] Punta M, Coggill P C, Eberhardt R Y, et al. The Pfam Protein Families Database[J]. Nucleic Acids Research, 2012, 40(1): 290-301.

[24] Loiselle S, Rouat J, Pressnitzer D, et al. Exploration of Rank Order Coding with Spiking Neural Networks for Speech Recognition[C]//Proc. of International Joint Conference on Neural Networks. [S. l.]: IEEE Press, 2005: 2076-2080.

[25] Xiong Tengke, Wang Shengrui, Jiang Qingshan, et al. A New Markov Model for Clustering Categorical Sequences[C]//Proc. of the 11th International Conference on Data Mining. [S. l.]: IEEE Press, 2011: 854-863.

[26] Kadouche R, Pigot H, Abdulrazak B, et al. User's Behavior Classification Model for Smart Houses Occupant Prediction[J]. Activity Recognition in Pervasive Intelligent Environments, 2011, 4(7): 149-164.

[27] Wei Dan, Jiang Qingshan, Wei Yanjie, et al. A Novel Hierarchical Clustering Algorithm for Gene Sequences[J]. BMC Bioinformatics, 2012, 13(1): 174-186.

编辑 顾逸斐

(上接第 174 页)

[8] Xue Mei, Ling Haibin. Robust Visual Tracking and Vehicle Classification via Sparse Representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(11): 2259-2272.

[9] Babenko B, Yang M H, Belongie S. Robust Object Tracking with Online Multiple Instance Learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1619-1632.

[10] Zhang Kaihua, Zhang Lei, Yang M H. Real-time Compressive Tracking[C]//Proc. of European Conference on Computer Vision. Florence, Italy: [s. n.], 2012: 866-879.

[11] Donoho D. Compressed Sensing[J]. IEEE Transactions on Information Theory, 2006, 52(4): 1289-1306.

[12] Candes E, Tao T. Near Optimal Signal Recovery from Random Projections and Universal Encoding Strategies[J]. IEEE Transactions on Information Theory, 2006, 52(12): 5406-5425.

[13] Candes E, Tao T. Decoding by Linear Programming[J]. IEEE Transactions on Information Theory, 2005, 51(12): 4203-4215.

编辑 顾逸斐