·人工智能及识别技术· 文章编号：1000-3428(2016)05-0201-06 文献标识码：A 中图分类号：TP18

基于符号熵的序列相似性度量方法

张 豪，陈黎飞，郭躬德

（福建师范大学数学与计算机科学学院福建省网络安全与密码技术重点实验室，福州 350007）

摘 要： 现有序列相似性度量算法在子序列相似性度量中仅考虑其局部相似度，忽略了其所属序列的整体结构信息。为此，提出一种以单个符号的熵为基础的序列相似性度量方法，根据同一序列中相同符号的位置及个数信息得出符号熵。通过凝聚型层次聚类结果验证序列相似性度量方法，在多个领域的符号序列数据集上的实验结果表明，与现有的基于子序列局部相似性方法相比，该相似性度量方法有效提高了聚类结果质量。

关键词： 符号序列；相似度；熵；层次聚类；序列聚类

中文引用格式： 张 豪，陈黎飞，郭躬德. 基于符号熵的序列相似性度量方法[J]. 计算机工程，2016，42(5)：201-206，212.

英文引用格式： Zhang Hao，Chen Lifei，Guo Gongde. Sequence Similarity Measurement Method Based on Symbol Entropy[J]. Computer Engineering，2016，42(5)：201-206，212.

Sequence Similarity Measurement Method Based on Symbol Entropy

ZHANG Hao，CHEN Lifei，GUO Gongde

（Fujian Province Key Laboratory of Network Security and Password Technology，
School of Mathematics and Computer Science，Fujian Normal University，Fuzhou 350007，China）

【Abstract】 Existing sequence similarity measurement algorithms only consider the local similarity of subsequences，ignoring global structure information. Thus，a similarity measurement method based on the entropy of single symbol for sequences is proposed. The entropy of a symbol is computed according to the positions and numbers of all the same symbols in a sequence. Through verifying the validity of the new sequence similarity measurement method by agglomerative hierarchical clustering，experimental results on a plurality of datasets show that，compared with the existing methods based on local similarity of substring，the new similarity measurement method can improve the clustering accuracy significantly.

【Key words】 symbol sequence；similarity；entropy；hierarchical clustering；sequence clustering

DOI： 10.3969/j.issn.1000-3428.2016.05.034

1 概述

符号序列是由取自有限符号集的若干个符号组成的符号串，在科研和商业领域普遍存在的序列有生物信息学领域的 DNA 序列、蛋白质序列以及语音识别领域的语音序列等[1-2]。数据挖掘的很多任务，如聚类、分类等都需要一个有效的相似性度量方法，因此，符号序列的相似性度量是分析和理解符号序列的关键任务之一。

迄今为止，研究者已提出多种相似性度量方法，其中大多数研究集中于数值型数据（或称连续型数据）相似性度量[3]。然而，不同于数值型数据[1]，符号序列不能使用欧氏距离等常用的相似性度量方法，而且不同序列间长度存在差异，属于同一序列的各个符号间存在局部的或全局的联系，使得符号序列相似性度量成为一项困难的任务。如何根据符号序列的整体结构信息与符号间的局部联系，有效地度量符号序列间相似度，是序列数据挖掘的关键问题，也是机器学习与数据挖掘领域富有挑战性的任务之一[2]。

当前符号序列相似性度量方法主要分为2 类[1]：基于马尔科夫模型的方法和基于子序列相似度的方法。基于马尔科夫模型的基本思想是把一个序列用一个参数化的随机过程来刻画[4]，不同簇的序列可以

基金项目： 国家自然科学面上基金资助项目“面向软件行为鉴别的事件序列挖掘方法研究”(61175123)；福建师范大学创新团队基金资助项目(IRTL1207)。

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

收稿日期： 2015-03-09 **修回日期：** 2015-04-27 **E-mail：** zhanghao_study@163.com

用不同参数的马尔科夫模型来代表。由此,序列间的相似度转化为序列与不同参数的马尔科夫模型之间的相似度[1]。然而,对于长度较短的序列和出现次数较少的子序列,这种基于统计的方法有效性将大大降低;此外,如何初始化代表不同簇的马尔科夫模型的参数是一个棘手问题[1]。基于子序列相似度的方法基于这样的思想[5]:两符号序列间相似或相同的子序列越多,其相似度越大。相比于基于马尔科夫模型的方法,基于子序列相似度的方法不存在参数初始化问题,且不受制于子序列出现次数;更重要的是,基于子序列相似度的方法可解释性更强,尤其是对于生物信息学领域的 DNA 序列、蛋白质序列。基于子序列相似度的方法需要应对如何确定子序列相似度的难题,现有的子序列相似性度量方法仅孤立地考虑了子序列局部相似度,而脱离了子序列所属的序列所包含的整体信息,这是不完备的。

聚类是机器学习与数据挖掘等领域研究的重点方向之一,其作为一种无监督学习方法是根据数据集本身特点使得相似度高的数据划分到同一簇,相似度低的数据划分到不同簇[6]。对数据集聚类可以探查数据和理解数据结构[4],因此,对序列数据集聚类是处理和分析序列的重要方法之一。为了有效地捕捉序列的整体信息,充分反映子序列与其所属序列的联系,避免目前主流方法中存在的子序列相似度丢失全局信息的问题,本文提出一种包含全局信息的子序列相似性度量方法,并给出了两序列的相似度度量方法,该方法可通过动态规划求出两序列的相似度。

2 相关工作

如前所述,符号序列是由非数值型的符号组成,符号间存在局部或全局的联系,序列间长度存在差异,这些特点使得符号序列的相似度度量是一个困难问题。如何有效地度量序列的相似性是符号序列聚类等需要解决的关键问题。下面介绍并分析现有的若干代表性方法。

基于马尔科夫模型算法的思想是:符号序列是一个由变长的符号串组成的马尔科夫过链[7],相邻的且又大于统计意义上的关联阈值的符号属于同一个符号串,否则属于不同的符号串。具体方法是按一定方法初始化马尔科夫模型的参数,用不同参数的马尔科夫模型代表不同簇的中心,通过计算各个马尔科夫模型生成一个序列的概率,把该序列划分到生成其概率最大的簇中,然后根据该序列更新该马尔科夫模型的参数。现有基于马尔科夫模型的代表性度量包括概率后缀树[7-10]等,CLUSEQ[11],DHCS[1]等算法都基于此模型对符号序列聚类。如前所述,对于长度较短的序列和在序列中出现频度较低但重要的子序列,基于统计的马尔科夫模型的有效性将大大降低,并且这类方法存在参数初始化选择问题[1]。所以,基于马尔科夫模型的算法在序列相似性度量时存在诸多难以解决的问题。

基于子序列相似度的基本思想是:两序列间相似或相同的符号子串越多,则相似性越高。基本方法是:(1)定义子序列相似度;(2)以子序列相似度为基础度量序列间相似度。然而,如何度量子序列间相似度而又不丢失序列整体结构信息是一个困难问题。如图 1 所示,S_1,S_2,S_3 为 3 个序列,设 s_1,s_2,s_3 分别为 S_1,S_2,S_3 的子序列,令 s_1 = CCAA;且 $s_2 = s_1$,$s_3 = s_1$。

S_1: SECCAAES
S_2: AACCAAQQ
S_3: NNCCAALL

图 1 3 个具有相同子序列的符号序列

现考虑 s_1,s_2,s_3 两两之间的相似度 $sim(s_1, s_2)$,$sim(s_1, s_3)$,$sim(s_2, s_3)$。

文献[12]提出如式(1)所示的方法,即两子序列的相似度为两者的最长公共子序列(LCS)长度与子序列长度比值,用文献[12]提出的子序列相似度计算方法得出:$sim(s_1, s_2) = 1$,$sim(s_1, s_3) = 1$,$sim(s_2, s_3) = 1$。

$$sim(x,y) = \frac{1}{n} \times |LCS(x,y)| \tag{1}$$

文献[13]提出如式(2)所示的方法,即若两子序列中对应位置的相同符号个数大于 l,子序列的相似度为 1,否则为 0。设式(2)中 $l = 1$,则根据文献[13]计算出:$sim(s_1, s_2) = 1$,$sim(s_1, s_3) = 1$,$sim(s_2, s_3) = 1$。

$$sim(x,y) = \begin{cases} 1 & \text{if } |x \wedge y| > l, |x \backslash y| < l \\ 0 & \text{else} \end{cases} \tag{2}$$

然而,直观考虑 s_1,s_2,s_3 中对应位置的符号“A”:在 S_1 与 S_3 中符号“A”分别在位置 5、位置 6 出现;在 S_2 中符号“A”分别在位置 1,2,5,6 出现,显然 s_1,s_2,s_3 中对应位置的符号“A”两两之间相似度必然与 S_1,S_2,S_3 中符号“A”出现的位置与个数信息有联系,此时定有 $sim(s_1, s_2) \neq sim(s_1, s_3)$;由于 S_1 与 S_3 中符号“A”,“C”在序列中位置一致,而且 S_1 与 S_3 两序列长度相同,从而可知 $sim(s_1, s_2) < sim(s_1, s_3)$;因为基于子序列相似度的方法是以子序列相似度为构件的,所以在 $sim(s_1, s_2) < sim(s_1, s_3)$ 情况下,有 $sim(S_1, S_2) < sim(S_1, S_3)$。然而,从式(1)与式(2)可以看出,子序列的相

似性度量序列与整体结构信息无关,这 2 种相似性度量显然是不完备的。

为了解决上述缺点,文献[14]提出子序列熵的方法。该方法把序列转换为 L^M 维向量,不同的维度对应不同子序列的熵,其中,L 表示子序列长度;M 表示符号集中元素个数。蛋白质与语音序列 M 分别为 20 与 26,显然这样得到的是高维数据,受“维度效应”的影响,高维数据聚类问题是目前数据挖掘的挑战性任务之一[3]。更重要的是,很多子序列在序列中仅出现一次,按照文献[14]给出的子序列熵的计算方法,该子序列的熵为 0,这显然是不合理的。

上述基于子序列相似度的方法仅考虑了子序列的局部相似性,而没有考虑序列中符号的个数与位置对子序列相似度的影响。虽然文献[14]克服了子序列相似度缺乏全局信息的问题,然而受“维度效应”影响,该方法并不适用于符号集中元素个数较多的数据集。针对这些问题,本文提出一种基于符号熵的子序列相似性度量方法(SES)。符号熵根据同一序列中相同符号的位置及个数信息而得出,因此,基于符号熵的子序列相似性度量有效地反映了序列的整体结构信息。本文在此基础上提出了基于动态规划思想的序列相似性度量方法。

3　新的相似度度量

本节详细阐述基于符号熵的序列相似性度量方法。首先给出符号熵的计算方法和子序列相似性度量方法,然后提出两序列相似性度量方法,最后给出新的相似性度量算法在凝聚型层次聚类中的应用。约定全文使用的记号如下:

定义 1　Σ 为符号集合(如 DNA 序列数据集对应的符号集合为{A,C,G,T};蛋白质序列数据集对应的符号集合为{A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y});$|\Sigma|$ 表示集合 Σ 中元素个数。

定义 2　设 x,y 为两符号序列,其中,$x = x_1x_2\cdots x_l$,$y = y_1y_2\cdots y_n$。对于 $\forall i$ 与 $\forall j$ 都有 $x_i \in \Sigma, y_j \in \Sigma$,$l$ 与 n 分别为序列 x 与 y 的长度,即序列中符号的个数为该序列的长度。

3.1　符号熵

结合图 1 示例所述,符号间的相似度与符号所在序列的位置及相同的符号在序列中出现的次数有关,即符号所携带的信息与序列的整体结构有关。香农熵[15]是测量一个信息源有序度的方法之一,可以定量反映信息源所携带的信息。文献[14]提出了 k-tuple(长度为 k 的符号串)的香农熵计算方法,在一定程度上反映了序列的结构信息。然而在一个序列中,尤其当 $|\Sigma|$ 较大时,大多数 k-tuple 出现次数少于或等于一次,这导致 k-tuple 熵为 0,从而使得 k-tuple熵计算失去意义。为了避免这个问题,转换文献[14]提出的计算 k-tuple 的熵方法为计算单符号的熵。

对于序列 x,p_r表示某个符号在 x 中第 r 次出现时的位置,计算 α_r:

$$\alpha_r = \frac{1}{p_r - p_{r-1}}, 1 \leqslant r \leqslant m \tag{3}$$

其中,m 表示该符号在 x 总共出现的次数 $p_0 = 0$;α_r 反映了该符号的密度且包含了该符号在符号出现的位置信息。

为了计算 α_r的有序性,定义 β_j 作为部分和,β_j 计算如下:

$$\beta_j = \sum_{r=1}^{j} \alpha_r, 1 \leqslant j \leqslant m \tag{4}$$

现根据式(4)构建一个离散概率分布 $Q = (q_1, q_2, \cdots, q_m)$,其中:

$$q_i = \beta_i / \sum_{j=1}^{m} \beta_j, 1 \leqslant i \leqslant m \tag{5}$$

然后计算 m 个位置的各个符号的香农熵,位于位置 p_i 的香农熵计算如下:

$$H(p_i) = -q_i \mathrm{lb}^{q_i}, 1 \leqslant i \leqslant m \tag{6}$$

可以看出,每个符号所对应的香农熵与该符号的位置与个数有关,有效地反映了序列的全局与局部信息。

3.2　SES 算法

下面介绍两子序列间相似度,并在此基础上介绍两序列间相似度计算方法。

现有的子序列相似性度量方法都是局部相似度,不能反映出序列的全局信息,现基于 3.1 节的符号熵提出下面的子序列相似性度量方法。

设 $s_1 = x_i x_{i+1}\cdots x_{i+k-1}$,$s_2 = y_j y_{j+1}\cdots y_{j+k-1}$ 分别为 x,y 的长度为 k(k 的取值将在实验部分讨论)的子序列,其中,$1 \leqslant i \leqslant l-k+1$,$1 \leqslant j \leqslant n-k+1$;设 s_1 与 s_2 的最长公共子序列公共 $LCS(s_1,s_2) = z_1z_2\cdots z_t$ $(0 \leqslant t \leqslant k)$;设对于任意一个 $z_r(1 \leqslant r \leqslant t)$,$s_1$ 与 s_2 中与之对应的符号分别是 p_{ir},p_{jr},则 $sim(s_1,s_2)$ 计算如下:

$$sim(s_1,s_2) = 1 - \frac{1}{k} \times \sum_{r=1}^{t} |(H(p_{ir}) - H(p_{jr}))| \tag{7}$$

由式(7)可以看出,子序列间相似度由对应位置的符号的熵之差组成,而符号的熵有效地反映了序列的局部与整体结构信息,避免了现有子序列度量

缺乏考虑序列整体结构信息的问题。

对于序列 $x=x_1x_2\cdots x_l$,$y=y_1y_2\cdots y_n$,令 $z_i=x_ix_{i+1}\cdots x_{i+k-1}$,$w_j=y_jy_{j+1}\cdots y_{j+k-1}$,把 x 与 y 扩展为:$x_l^+=z_1z_2\cdots z_{m-k+1}$,$y_n^+=w_1w_2\cdots w_{m-k+1}$,在不考虑两序列长度影响的情况下,$x_l^+$ 与 y_n^+ 的相似度可用下式迭代地求出:

$$sim(x_l^+,y_n^+)=\max(sim(x_{l-1}^+,y_n^+),sim(x_l^+,y_{n-1}^+),sim(x_{l-1}^+,y_{n-1}^+)+sim(z_{l-k+1},w_{m-k+1})) \tag{8}$$

其中,$sim(z_{l-k+1},w_{m-k+1})$ 的计算即是式(7)中两子序列的相似性度量计算方法。

为了降低序列长度给序列相似性度量带来的偏倚,采取的方法如下:

$$sim(x,y)=sim(x_l^+,y_n^+)/\max(l,n)$$

新的相似性度量方法的算法描述如下:

算法 1 SES 方法

输入 $k,x=x_1x_2\cdots x_l,y=y_1y_2\cdots y_n$

输出 序列 x 与序列 y 的相似度 $sim(x,y)$

```
(1) for i = 1 to l
        Set sim(x_{i+},y_{0+}) = 0;
    for j = 1 to n
        Set sim(x_{0+},y_{j+}) = 0;
(2) for i = 1 to l
        for j = 1 to n
            sim(x_{i+},y_{j+}) = max(sim(x_{i-1+},y_{j+}),sim(x_{i+},y_{j-1+}),sim(x_{i-1+},y_{j-1+}) + sim(z_{i-k+1},w_{j-k+1}))
(3) sim(x,y) = sim(x_{l+},y_{n+})/max(l,n)
```

算法的第(1)步时间复杂度为 $O(n+l)$;在算法第(2)步中,每次计算两序列的两子序列相似度需要计算两子序列的 *LCS*(最长公共子序列)与 *LCS* 中各个对应符号的熵,若按照经典的 *LCS* 解法其时间复杂度为 $O(k^2)$,然而根据文献[16],其时间复杂度可以降为 $O(k\text{lb}k)$,计算一个符号的熵(式(6)),其时间复杂度为 $O(m)$(此处假设该符号在该序列中共出现 m 次);所以计算两序列相似度的总的时间复杂度约为 $O(nlk\text{lb}k+(m+1)(l+n))$。

根据算法 1,图 1 中三子序列的相似度为:$sim(s_1,s_2)=sim(s_2,s_2)\approx0.951$,$sim(s_1,s_3)=1$;三序列的相似度为:$sim(S_1,S_2)=sim(S_2,S_3)\approx0.109$,$sim(S_1,S_3)=0.125$。因此有 $sim(s_1,s_2)<sim(s_1,s_3)$,$sim(S_1,S_2)<sim(S_1,S_3)$,从而与图 1 中三序列之间相似度大小关系的预期结果一致,说明了新方法有效地克服了现有方法中子序列度量缺乏序列整体结构信息的问题。

3.3 基于符号熵度量的序列聚类

符号序列聚类通常需要合适的序列相似性度量方法和合适的聚类算法[2]。在同一种聚类算法下,应用不同相似性度量方法的聚类结果通常会不同,因此,可以根据聚类结果来衡量不同相似度量算法的有效性。在诸多聚类算法中,层次聚类是适用于序列两两相似度的常用的聚类算法[2]。

层次聚类分为凝聚型和分裂型,其中以凝聚型层次聚类最受青睐[2]。凝聚型层次聚类从只包含一个样本的 N 个簇开始,重复合并相似的簇直到只有一个簇。有 3 种方法应用于凝聚型层次聚类中簇间相似性度量:单链接,全链接和平均链接[2,4,6]。相比于全链接和平均链接,单链接方法具有不受簇合并顺序影响和能发现任意形状簇等优点[2,4]。

下面将给出序列的相似性度量在凝聚型层次聚类中应用的算法及时间复杂度分析。

算法 2 基于 SES 的凝聚层次聚类

输入 样本集 S

输出 层次聚类树

步骤 1 根据 3.2 节中两序列相似度计算方法求出 S 中的每 2 个样本间的相似度。

步骤 2 用 C 表示各个簇的集合。初始化 $C=\{$所有样本$\}$。

步骤 3 选取 C 中属于不同簇的 2 个样本 p_1,p_2,使得 $SIM(p_1,p_2)=\max\{sim(p_3,p_4)\mid p_3\in C,p_4\in C\}$,合并 p_1 与 p_2 所在的簇为一个簇。

步骤 4 若 C 仅包含一个簇,算法结束;否则,转到步骤 3。

对于 N 个样本,需要计算 $N(N-1)$ 个样本间的相似度,故算法 2 中步骤 1 的时间复杂度为 $O(N^2(nlk\text{lb}k+m(l+n)))$;在步骤 3 ~ 步骤 4 中,使用快速排序法对样本间相似度按从小到大顺序排序,时间复杂度是 $O(N^2\text{lb}N)$;所以算法的总时间复杂度约为 $O(N^2\text{lb}N)$。

4 实验与结果分析

4.1 实验数据集

为了验证算法的有效性,选取基因、语音和蛋白质这 3 种不同领域的序列样本集。实验使用了 5 个数据集,分别用 DS_1,DS_2,DS_3,DS_4,DS_5 表示。DS_1 来自于 PBIL(http://pbil.univ-lyon1.fr/)的同源脊椎动物基因数据库 HOVERGEN[14];DS_2 来自于 PBIL 的同源微生物基因数据库 HOMOLENS[14];DS_3 来自于 PBIL 的同源微生有机物基因数据库 HOGENOM[14]。由于 PBIL 在不断更新,因此 DS_1 和 DS_2 样本数比文献[14]有所增加。DS_4,DS_5 分别是 5 个单独的法语元音('a','e','i','o','u'),它经常以序列的形式应用于语音识别[1];DS_6 是传统的生物序列比对方法很难区分的"(a/b)8-barrel"

蛋白质序列[5]。各数据集的详细参数如表 1 所示,其中,*NS* 为数据集中序列个数;*AEL* 为该数据集中序列的平均长度;*NC* 为数据集中序列类别数目;*NCN* 为数据集中各类别包含的样本数。

表 1　实验数据集的相关信息

DB	数据集描述	*NS*	*AEL*	*NC*	*NCN*
DS_1	Homologous vertebrate genes	285	1 307	6	23:50:26:35:92:59
DS_2	homologous Ensembl genes	251	1 074	6	29:28:55:70:21:48
DS_3	homologous organisms genes	310	1 535	6	32:34:65:32:99:48
DS_4	speech sequences	50	1 898	5	10:10:10:10:10
DS_5	speech sequences	50	1 603	5	10:10:10:10:10
DS_6	(a/b)8-barrel protein sequences	33	871	5	6:7:6:8:6

4.2　对比算法与评价指标

为了测试新的序列比对算法的性能,选用 N-gram[12] 和 SCS[13] 这 2 个主流的序列相似性度量方法作为对比算法;选取基于单链接的凝聚型层次聚类算法的聚类结果衡量序列相似性度量方法的有效性。为了进一步测试序列相似度度量性能,另选用文献[14]中的序列相似性度量方法 DMk 与 CLUSEQ[11] 这个基于 PST[7-10] 的序列聚类算法作为比对算法。

常用的聚类有效性评价方法有外部评价法、内部评价法和相对评价法[17]。由于实验数据的实际类别已知,这里采用常用的聚类有效性评价外部指标 *F*-measure[18] 来评价聚类结果的有效性。对于实际类别 i 和聚类 j,评价指标 $F(i,j)$ 定义为:

$$F(i,j) = \frac{2 \times prcision(i,j) \times recall(i,j)}{prcision(i,j) + recall(i,j)} \quad (9)$$

其中,$i=1,2,\cdots,e$;$j=1,2,\cdots,k$;$precision(i,j)=n_{ij}/n_j$;$recall(i,j)=n_{ij}/n_i$;$e$ 是实际类别数目;k 是聚类数目;n_{ij} 是聚类 j 中属于实际类别 i 的样本的数目;n_i 是实际类别 i 中样本的数目;n_j 是聚类 j 中样本的数目。聚类结果的总体 F-*measure* 定义如下:

$$F = \sum_i \frac{n_i}{N} \max(F(i,j)) \quad (10)$$

其中,N 是样本集样本数目。

4.3　实验设置

在序列的相似性度量中,子序列长度值采用该领域相似性度量常用的值。根据文献[14],DNA 序列其子序列的长度 k 取值为 3;根据文献[1,13],蛋白质序列与语音序列其子序列长度 k 取值为式(11)中的 $K_{X,Y}$。

$$K_{X,Y} = \frac{\log_a(|X|^2 + |Y|^2) + \log_a \lambda_{X,Y}(1-\lambda_{X,Y}) + 0.57}{-\log_a \lambda_{X,Y}} \quad (11)$$

$$\lambda_{X,Y} = \max\left(\sum_{i=1}^{m} (p_i^X)^2, \sum_{i=1}^{m} (p_i^Y)^2\right) \quad (12)$$

其中,p_i^X 和 p_i^Y 是第 i 个类别的符号在序列 X 和 Y 中出现的频率。

实验中对层次聚类树划分时,选取合适的阈值,使得聚类数目等于实际类别数。实验中的各个算法分别在每个数据集上进行了 5 次测试,取 5 次聚类精度的平均值。

4.4　结果分析

从表 2 可以看出,在 5 次实验中,基于单链接的凝聚聚类精度的方差为 0,这是因为数据集中数据量较少时,基于单链接的凝聚聚类过程是比较稳定的;而 CLUSEQ[11] 是随机初始化各簇中心,所以聚类精度的方差较大。相比于 N-gram[12],SCS[13],DMk[14],CLUSEQ[11],新的序列相似性度量方法表现良好,验证了使用基于符号熵的子序列相似性度量可以有效地提高子序列相似性度量的准确度,从而提高序列相似性度量的有效性。与表 2 中的 3 个序列比对算法的聚类精度相比,CLUSEQ[11] 的聚类精度较低,说明在序列聚类中使用基于子序列相似度度量的方法比使用统计模型方法更合理。

表 2　基于不同聚类方法各个序列相似度算法的聚类精度

DB	SES 方法	N-gram 方法	SCS 方法	DMk 方法	CLUSEQ 方法
DS_1	0.843 4 ± 0.000 0	0.763 0 ± 0.000 0	0.760 8 ± 0.000 0	0.873 1 ± 0.000 0	0.465 6 ± 0.203 5
DS_2	0.863 1 ± 0.000 0	0.706 5 ± 0.000 0	0.335 1 ± 0.000 0	0.840 3 ± 0.000 0	0.360 1 ± 0.157 2
DS_3	0.901 3 ± 0.000 0	0.901 3 ± 0.000 0	0.706 3 ± 0.000 0	0.886 5 ± 0.000 0	0.547 3 ± 0.102 9
DS_4	0.937 7 ± 0.000 0	0.848 1 ± 0.000 0	0.842 1 ± 0.000 0	0.475 9 ± 0.000 0	0.609 5 ± 0.095 7
DS_5	1.000 0 ± 0.000 0	0.844 5 ± 0.000 0	0.851 3 ± 0.000 0	0.338 5 ± 0.000 0	0.682 1 ± 0.083 5
DS_6	0.856 1 ± 0.000 0	0.824 8 ± 0.000 0	0.833 0 ± 0.000 0	0.450 2 ± 0.000 0	0.492 7 ± 0.263 3

CLUSEQ[11]采取的随机选取聚类中心的方法会影响聚类精度,此外,序列的重要片段并不一定具有较高的统计频率,所以以 PST[7-10]模型为基础的 CLUSEQ[11]并不可靠。N-gram[12]在子序列相似度度量中采用最长公共子序列的方法,显然其认为公共子序列中的一个符号所对应的两子序列中的符号是完全等价的,并没有考虑该符号在整个序列中的位置及个数信息。SCS[13]是以两子序列中匹配符号的个数与临界值大小来判断两子序列相似是 0 还是 1,显然这是定性的方法且没有考虑符号在整个序列中的个数及位置信息。因此,以子序类相似度度量为基础的 N-gram[12]与 SCS[13]由于是以子序列间局部相似度为基础的,在子序列相似度度量中丢失了序列的整体结构信息。相比于在语音与蛋白质上的聚类精度,DMk[14]在 3 个 DNA 序列集上表现较好,这是因为在符号集中元素较多时,该方法提取出大量无效特征,且很多仅出现一次的子序列以该方法计算出的熵为 0,导致其有效性大大降低。

以符号熵为基础的序列相似性度量方法,通过符号的位置及个数信息引入符号的密度,进而计算出符号香农熵;子序列相似性度量通过计算符号间的熵差来定量度量子序列的相似度。实验结果表明,该相似性度量算法有效地处理了局部信息与全局信息之间的矛盾。

下面测试新方法的可伸缩性。为检验序列长度对相似性度量计算时间的影响,对含有 AGCT 4 个符号的合成的随机序列集进行测试,测试结果如图 2 所示。测试方法如下:选用 N-gram[12]与 SCS[13]作为对比方法;SES 生成 6 个随机序列集 1 ~ 6,每个序列集包含 10 个样本,其中序列集 i 中每个随机序列的长度为 1 000i;记录上述各个算法在每个数据集上运行 5 次的度量两样本间相似度花费的平均时间。设 k 为各个算法中匹配的子序列长度值,L_1,L_2 为两序列的长度;根据文献[12],N-gram[12]的计算复杂度为 $O(k^2L_1L_2)$;根据文献[13],SCS[13]的计算复杂度为 $O(\sqrt{k\mathrm{lb}k}\ L_1L_2)$;SES 的计算复杂度为 $O(nlk\mathrm{lb}k+(m+1)(l+n))$,因为 $(m+1)(l+n) << nlk\mathrm{lb}k$,所以 SES 的计算复杂度约为 $O(nlk\mathrm{lb}k)$。可以看出,这 3 个算法时间复杂度都为 $O(aL_1L_2)$ 形式(其中,a 具体取值与具体算法有关),所以随着序列长度的变化,其时间开销呈二次函数型增长。由图 2 可以看出,N-gram 时间开销最多,SES 与 SCS 时间开销差距较小,而理论计算复杂度之间关系满足 $k^2L_1L_2 > k\mathrm{lb}k\ L_1L_2 > \sqrt{k\mathrm{lb}k}\ L_1L_2$ 且 $(k^2-k\mathrm{lb}k)L_1L_2>(k\mathrm{lb}k-\sqrt{k\mathrm{lb}k})\ L_1L_2$,所以,图 2 表明了实验中的时间开销与理论的时间开销相吻合,而且可以看出,序列长度较短时,时间开销与序列长度接近线性关系,此时 SES 的时间开销在可接受的范围内。

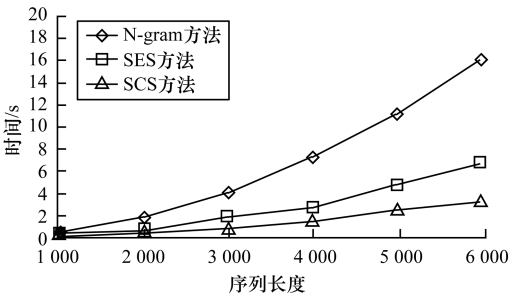

图 2 3 个相似性度量方法的时间开销

5 结束语

现有的基于子序列相似度的序列相似性度量算法在子序列相似性度量中仅考虑了其局部相似度,忽略了其所属序列的整体结构信息,针对此问题,本文提出了新的相似性度量算法。在相似性度量算法中,给出基于符号熵的子序列相似性度量方法,有效地克服了子序列度量中局部相似度丢失整体结构信息的问题。实验结果表明,与现有序列相似性度量算法相比,SES 在多个领域的符号序列数据集上的聚类应用中有较好的聚类结果。下一步工作将研究不同符号间相对位置、个数所蕴含的信息及这些信息的有序化。

参考文献

[1] Xiong T, Wang S, Jiang Q, et al. A New Markov Model for Clustering Categorical Sequences[C]//Proceedings of International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2011: 854-863.

[2] Dong Guozhu, Pei Jian. Sequence Data Mining[M]. New York, USA: Springer-Verlag New York Inc., 2007.

[3] 陈黎飞,郭躬德. 属性加权的类属型数据非模聚类[J]. 软件学报,2013,24(11):2628-2641.

[4] Alpaydin E. 机器学习导论[M]. 范 明,译. 北京:机械工业出版社,2009.

[5] Kelil A, Wang S, Brzezinski R, et al. CLUSS: Clustering of Protein Sequences Based on a New Similarity Measure[J]. BMC Bioinformatics, 2007, 8(1): 286.

[6] 孙吉贵,刘 杰,赵连宇. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.

[7] Ron D, Singer Y, Tishby N. ThePower of Amnesia: Learning Probabilistic Automata with Variable Memory Length[J]. Machine Learning, 1996, 25(2-3): 117-149.

[8] Grossi R, Vitter J. Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching[C]//Proceedings of ACM STOC'00. New York, USA: ACM Press, 2000: 397-406.

[9] Gusfield D. Algorithms on Strings, Trees, and Sequences[J]. ACM SIGACT News, 1997, 28(4): 41-60.

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

(下转第 212 页)

6 结束语

池化层是卷积神经网络模型中非常重要的结构,为了提升池化层的工作效率,本文提出 DA-Pooling 算法。该算法考虑到语音数据具有局部相关的特性,根据局部相似性度量对 Mean-Pooling,Max-Pooling 和 Stochastic-Pooling 算法进行有效整合,增强了 Pooling 算法对语音数据的适应性。实验结果表明,DA-Pooling 算法相比其他 Pooling 算法,在识别准确率和可扩展性方面具有较好性能。下一步工作将加入沿时间轴的 Pooling 算法以降低语音数据嘈杂度,从而达到更好的识别效果。

参考文献

[1] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks[J]. Science, 2006, 313(5786): 504-507.

[2] Sinharoy B, van Norstrand J A, Eickemeyer R J, et al. IBM POWER8 Processor Core Microarchitecture[J]. IBM Journal of Research and Development, 2015, 59(1): 1-21.

[3] Abdel-Hamid O, Mohamed A, Jiang H, et al. Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition[C]//Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2012: 4277-4280.

[4] Sainath T N, Mohamed A, Kingsbury B, et al. Deep Convolutional Neural Networks for LVCSR[C]//Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2013: 8614-8618.

[5] Vanhoucke V, Senior A, Mao M Z. Improving the Speed of Neural Networks on CPUs[C]//Proceedings of Deep Learning and Unsupervised Feature Learning NIPS Workshop. [S. l.]: NIPS, 2011: 1-8.

[6] Dean J, Corrado G, Monga R, et al. Large Scale Distributed Deep Networks[Z]. 2012.

[7] 张佳康,陈庆奎. 基于 CUDA 技术的卷积神经网络识算法[J]. 计算机工程,2010,36(15):179-181.

[8] Sainath T N, Kingsbury B, Mohamed A, et al. Improvements to Deep Convolutional Neural Networks for LVCSR[C]//Proceedings of International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2013: 315-320.

[9] Boureau Y L, Ponce J, Lecun Y. A Theoretical Analysis of Feature Pooling in Visual Recognition[C]//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel: IMLS Press, 2010: 111-118.

[10] Zeiler M D. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks[EB/OL]. [2015-07-16]. http://www.arxiv.org/pdf/1301.3557.pdf.

[11] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9): 1904-1916.

[12] Lehmann E L, Abrera H J M. Nonparametrics: Statistical Methods Based on Ranks[J]. International Encyclopedia of Education, 2010, 83(1977): 347-353.

[13] 张晴晴,刘 勇,潘接林,等. 基于卷积神经网络的连续语音识别[J]. 工程科学学报,2015,(9):1212-1217.

[14] Jia Y, Shelhamer E, Donahue J, et al. CAFFE: Convolutional Architecture for Fast Feature Embedding[C]//Proceedings of ACM International Conference on Multimedia. New York, USA: ACM Press, 2014: 675-678.

[15] Panayotov V, Chen G, Povey D, et al. Librispeech: An ASR Corpus Based on Public Domain Audio Books[C]//Proceedings of International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2015: 5206-5210.

编辑 陆燕菲

(上接第 206 页)

[11] Yang J, Wang W. CLUSEQ: Efficient and Effective Sequence Clustering[C]//Proceedings of IEEE International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2003: 101-112.

[12] Kondrak G. N-gram Similarity and Distance[C]//Proceedings of IEEE International Conference on String Processing and Information Retrieval. Washington D. C., USA: IEEE Press, 2005: 115-126.

[13] Kelil A, Wang S. SCS: A New Similarity Measure for Categorical Sequences[C]//Proceedings of IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2008: 343-352.

[14] Wei D, Jiang Q, Wei Y, et al. A Novel Hierarchical Clustering Algorithm for Gene Sequences[J]. BMC Bioinformatics, 2012, 13(1): 174.

[15] Schmitt A O, Herzel H. Estimating the Entropy of DNA Sequences[J]. Journal of Theoretical Biology, 1997, 188(3): 369-377.

[16] Longest Common Subsequence. [EB/OL]. (2012-10-21). http://www.cs.ucf.edu/courses/cap5937/fall2004/Longest%20common%20subsequence.pdf.

[17] Halkidi M, Batistakis Y, Vazrgiannis M. On Clustering Validation Techniques[J]. Intelligent in Formation Systems, 2001, 17(2-3): 107-145.

[18] Larsen B, Aone C. Fast and Effective Text Mining Using Linear-time Document Clustering[C]//Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 1999: 16-22.

编辑 索书志