文章编号:1007-1423(2018)33-0025-04 DOI:10.3969/j.issn.1007-1423.2018.33.006

基于隐马尔可夫模型的中文分词

吴帅,潘海珍

(上饶师范学院数学与计算机科学学院,上饶 334001)

摘要:
中文分词是搜索引擎、机器翻译、情感分析等自然语言处理的基础,分词的准确率和效率对后续的工作有着非常大的影响。目前性能比较好的分词算法是基于统计机器学习的方法,隐马尔可夫模型能够较好地描述词与词之间的前后关系。论述模型实现中文分词的基本原理,并给出模型的 Python 实现。

关键词:
隐马尔可夫模型; 中文分词; 分词算法; Python

0 引言

中文分词是中文自然语言处理中的基础环节,由于中文的词语之间没有明显的分隔符,使得中文相对其他语言的分词难度更大,中文分词的质量和分词效率将会影响建立在其基础上的高级应用。中文分词也是中文自然语言处理中的瓶颈问题,解决好了中文分词,将会给其他相关领域的研究带来突破性的发展。中文分词的研究工作已经持续了三十多年,分词的准确度和速度得到非常大的提高,目前比较流行且效果比较好的方法是基于统计的机器学习方法。

隐马尔可夫模型是一种基于统计的机器学习模型,可以通过观测数据来预测数据最可能的原始状态,这点正好满足中文分词的要求,将汉字序列切分成一个个独立且最合理的词。首先为中文文本建立统计模型,利用隐马尔可夫假设简化模型,降低计算的复杂度,最后通过 Viterbi 算法来预测最佳的词方式。本文分析了隐马尔可夫模型实现中文分词的基本原理、过程及分词模型的 Python 实现。

1 中文分词算法

近年来,专家学者们提出了许多的中文分词算法,可以归纳为三大类:基于词典匹配的算法、基于统计的算法和基于理解的算法。基于词典匹配的算法是按照一定的策略将文本中准备分析的字符串与词典中的词语进行匹配,若在词典中找到某个字符串,则识别出文本中的词语;基于统计的算法是基于人的直观理解,任意相邻的汉字出现的频率越高,说明它们组成词的可能性就越大;基于理解的算法是让计算机模拟人对句子的理解,达到识别词的效果,分词的同时进行句法、语义分析,利用句法信息和语义信息来处理歧义现象,该方法还处于理论研究,未有实际的应用。中文分词中存在歧义识别和未登录词识别两大难点。各类算法的具体比较见表 1。

表 1 三类分词算法的比较

算法类型 / 性能/要求	基于词典	基于统计	基于理解
分词速度	较快	一般	慢
分词准确度	一般	较准	准
算法复杂度	容易	一般	难
技术成熟度	成熟	成熟	不成熟
未登录词识别能力	差	较好	好
歧义处理能力	差	较好	好
是否需要词典	是	否	否
是否需要语料库	否	是	否
是否需要语言规则	否	否	是

2 隐马尔可夫模型概述

2.1 隐马尔可夫模型

隐马尔可夫模型(Hidden Markov Model,HMM)描

述了含有隐藏变量的马尔可夫随机过程，该模型涉及两个序列和三个概率矩阵，即可观察的观测序列 O、隐藏的状态序列 Z、初始的状态概率矩阵π，状态转移概率矩阵 A 及状态生成观测的概率矩阵 B。HMM 可表示为{O,Z,π,A,B $\lambda=(\pi,A,B)$}，λ=(π,A,B)决定 HMM 模型，PI π 和 A 决定观测序列，B 决定状态序列。

HMM 具有三个基本问题：概率计算问题、学习问题和预测问题。概率计算问题是计算在模型λ下观测序列 O 的概率 P(O/λ)，直接求解的方法不可行，计算量非常大，有效的方法是前向-后向算法。学习问题是已知观测 O 估计模型 $\lambda=(\pi,A,B)$ λ=(π,A,B)的参数，有监督可用极大似然估计法、无监督可用 Baum-Welch 算法。预测问题是给定观测序列，求出最有可能的对应的状态序列，可用近似算法和 Viterbi 算法。

2.2 应用原理

中文的词是由汉字构成，每个汉字在构词时都有一个确定的位置。字在词中出现的位置可用 BMSE 四种标签表示，B 表示词的开始位置、M 表示多字词的中间位置、E 表示词的结束位置，S 表示字单独成词。如“明月湖的荷花露出迷人的笑脸”对应的词位标签序列为“BMESBEBESBE”，分词结果为“明月湖/的/荷花/露出/迷人/的/笑脸”。

文本中的每个字构成观测序列，每个字的词位标注构成状态序列。中文分词就转换为求解字的词位标注问题，基于已加工好的语料库训，训练得到 HMM 的参数 $\lambda=(\pi,A,B)$，再通过 Viterbi 算法得到待分词文本的词位标注序列，从而得到最佳分词。

3 HMM在中文分词中的实现

隐马尔可夫模型实现中文分词主要由三个步骤组成，即训练、预测和分词，如图 1 所示。

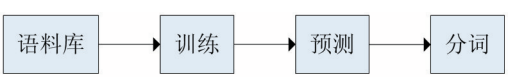

图1 HMM的中文分词过程

(1)训练

通过统计语料库中相关信息训练 HMM 中的三个参数 PI、A 和 B。A 表示字的词位状态转移矩阵，B 表示词位到词的混淆矩阵。从语料库中可以获得每个词位出现的次数，每个字符出现的次数，通过频率代替概率得到三个参数的值。

$$A_{ij}=P(Z_j/Z_i)=\frac{P(Z_i,Z_j)}{P(Z_i)}\approx\frac{freq(Z_i,Z_j)}{freq(Z_i)} \quad (1)$$

$$B_{ij}=P(O_j/Z_i)=\frac{P(O_j,Z_i)}{P(Z_i)}\approx\frac{freq(O_j,Z_i)}{freq(Z_i)} \quad (2)$$

公式中 Z={B,M,E,S}为字的词位序列，O={字符集}为观测序列，$freq(Z_i,Z_j)$表示 Z_iZ_j 在语料库中相邻同时出现的次数，$freq(O_i,Z_i)$表示字符 O_i 和 Z_i 某个词位同时出现的次数。HMM 在分词中的状态转移概率矩阵为：

$$A=\begin{bmatrix}0.0 & p(M/B) & p(E/B) & 0.0\\ 0.0 & p(M/M) & p(E/M) & 0.0\\ p(B/E) & 0.0 & 0.0 & p(S/E)\\ p(B/S) & 0.0 & 0.0 & p(S/S)\end{bmatrix}$$

计算过程中会出现频数为零或很小的值，为了避免出现计算结果的下溢，对频数取对数，如 $A_{ij}=\log(freq(Z_iZ_j)-\log(Z_i))$ $A_{ij}=\log(freq(Z_iZ_j))-\log(freq(Z_i))$。程序中采用的是北京大学加工的 1998 年《人民日报》语料库，该语料库具有较为完整的加工规范说明，目前较为成熟，被研究人员普遍采用。

若训练样本的数据不足，混淆矩阵 B 会过于稀疏。矩阵 B 的形式为：

$$B=\begin{bmatrix}p(o_1/B) & p(o_2/B) & \cdots & p(o_n/B)\\ p(o_1/M) & p(o_2/M) & \cdots & p(o_n/M)\\ p(o_1/E) & p(o_2/E) & \cdots & p(o_n/E)\\ p(o_1/S) & p(o_2/S) & \cdots & p(o_n/S)\end{bmatrix}$$

给定观测序列学习 HMM 模型参数，采用 Baum-Welch 算法[4]训练分词模型，参数估计公式分别为：

$$\gamma_t=\alpha_t(i)\beta_t(i)/\sum_{j=1}^{N}\alpha_t(j)\beta_t(j),$$

$$\varepsilon_t(i,j)=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum\sum\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

$$a_{ij}=\sum_{t=1}^{T-1}\varepsilon_t(i,j)/\sum_{t=1}^{T-1}\gamma_t(i),$$

$$b_j=\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)/\sum_{t=1}^{T}\gamma_t(i),\ \pi_i=\gamma_1(i)。$$

算法的实现代码如下：

```
def Baum_Welch(PI, A, B):
    f = file(".\\observed.txt")
    st= f.read()[3:].dc('utf-8') # 获取观测序列
    f.close()
```

```
    T = len(st)
    alpha = [[0 for i in range(4)] for t in range(T)]
    beta = [[0 for i in range(4)] for t in range(T)]
    gamma = [[0 for i in range(4)] for t in range(T)]
    ksi = [[[0 for j in range(4)] for i in range(4)]
    for t in range(T-1)]
        for time in range(100):
            print("time:", time)
            calc_alpha(PI, A, B, st, alpha) #求α
            calc_beta(PI, A, B, st, beta) #求β
            calc_gamma(alpha, beta, gamma)#求γ
            calc_ksi(alpha, beta, A, B, st, ksi)#ε
            bw(PI, A, B, alpha, beta, gamma, ksi, st)
            save_parameter(PI, A, B, time)#模型参数
```

（2）预测

从语料库中训练 HMM 分词模型后，可通过 Viterbi 算法来预测未知语言中汉字的词位标记从而达到分词的目的，可以求得全局最优的分词结果。Viterbi 算法实际是用动态规划来求解隐马尔可夫模型预测问题，即用动态规划求概率最大路径。δ_t 定义在时刻 t 状态为 i 的所有单个路径$(i_1,i_2,\cdots,i_t)$中的概率最大值为：

$$\delta_t = \max_{i_1,i_2,\cdots,i_{t-1}} P(i_t = i, i_{t-1},\cdots,i_1,o_1|\lambda)$$

递推公式为：

$$\delta_{t+1} = \max_{1\leqslant j\leqslant N}[\delta_t(j)a_{ij}]b_i(o_{t+1})$$

定义在 t 状态为 i 的所有单个路径中概率最大的路径的第 t−1 个结点为：

$$\psi_t = \arg\max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}], i = 1,2,\cdots,N$$

Viterbi 算法的实现如下：

```
def Viterbi(PI, A, B,O):
    T = len(O)   # 观测序列
    delta = [[0 for i in range(4)] for t in range(T)]
    pr = [[0 for i in range(4)] for t in range(T)]
    for i in range(4):
        delta[0][i] = PI[i] + B[i][ord(O[0])]
    for t in range(1, T):
        for i in range(4):
            delta[t][i] = delta[t-1][0] +A[0][i]
        for j in range(1,4):
            xj = delta[t-1][j] + A[j][i]
            if delta[t][i] < xj:
            delta[t][i] = xj
            pr[t][i] = j
        delta[t][i] += B[i][ord(O[t])]
    sq = [-1 for t in range(T)]
    k = 0
    for i in range(1, 4):#0B1M2E3S
        if delta[T-1][i] > delta[T-1][k]:
            k = i
        sq[T-1] = k
        for t in range(T-2, -1, -1):
            k = pr[t+1][k]
            sq [t] = k
    return sq
```

（3）分词

一般情况下，完成文本的标注序列后，需要进行分词，分词的方法是从左到右，采用最大匹配模式。程序中分词的实现如下所示。

```
def segment(st, sq):
    N = len(st)
    i = 0
    while i < N:  #B/M/E/S
        if sq[i] == 0 or sq[i] == 1:# B
            j = i+1
            while j < N:
                if sq[j] == 2:
                    break
                j += 1
            print(st[i:j+1], '/', end=' ')
            i = j+1
        elif  sq [i] == 3 or sq[i] == 2:  #S
            print(st[i:i+1], '/', end=' ')
            i += 1
        else:
            print('错误:', i, sq[i])
```

基于 HMM 的分词模型经常会将一起出现频率高的字组切分成词，如“我的”、“每个”等，会出现错误分词的现象。有训练语料时，训练模型的时间较短。HMM 是生成模型，即使没有先验语料，也可以使用 EM 方法进行估计，估计原则是使每个序列的 P(X)最大，这个优势是判别模型无法比拟的。

4　结语

本文分析隐马尔可夫模型的理论基础，论述基于

隐马尔可夫模实现中文分词的基本原理。HMM 分词模型只考虑词前后关系，未考虑词的上下文之间的关系，但在中文分词中表现较好，HMM 可以求得全局最优的分词结果。中文分词涉及的范围非常广，由于中文本身的特殊性，中文分词算法在不断地发展和完善，在分词速度更快、精度更高、歧义词、未登录词、新词的识别等方面会得到突破。

参考文献：

[1]魏光泽. 中文分词技术在搜索引擎中的研究与应用[D]. 青岛:青岛科技大学,2016.

[2]奉国和,郑伟. 国内中文自动分词技术研究综述[J]. 图书情报工作,2011.1.

[3]李航. 统计学习方法. 清华大学出版社[M],2012.

[4]周志华. 机器学习. 清华大学出版社[M],2016.

[5]冯雪. 中文分词模型词典融入方法比较[J]. 计算机应用研究,2018.2.

[6]王庆福. 隐马尔可夫模型在中文文本分词中应用研究[J]. 无线互联网科技,2016

[7]Andreas C.Müller,Sarah Guido,著. Python 机器学习基础教程. 张亮,译. 人民邮电出版社[M],2018.

作者简介：

吴帅(1980-)，女，江西玉山人，硕士，研究方向为机器学习、人工智能

收稿日期：2018-09-20　　修稿日期：2018-10-15

Chinese Word Segmentation Based on Hidden Markov Mode

WU Shuai, PAN Hai-zhen

(School of Mathematics and Computer Science, Shangrao Normal University, Shangrao 334001)

Abstract：

Chinese Word Segmentation is the basis of Natural Language Processing such as search engine, machine translation, emotional analysis, etc. The accuracy and efficiency of word segmentation have a great impact on subsequent work. The current segmentation algorithm with better performance is based on statistical machine learning, Hidden Markov Model can better describe the relationship between words. Discusses the basic principle of Chinese Word Segmentation based on HMM, and presents the Python implementation of the model.

Keywords：

Hidden Markov Model; Chinese Word Segmentation; Word Segmentation Algorithm; Python