文章编号: 2096-1618(2018)01-0034-05

基于变长隐马尔科夫模型的维基词条编辑微过程挖掘

黄冠英， 郑皎凌
(成都信息工程大学软件工程学院,四川 成都 610225)

摘要: 建立一种基于变长隐马尔科夫模型的维基词条编辑微过程挖掘方法。由于传统的 EM 算法需要指定隐藏状态的数目,而隐状态数目通常需要通过对实际数据的大量人工观察得到,这就使隐状态数目的设置具有较大的主观性。新方法首先基于张量分解来挖掘维基词条编辑微过程的隐状态数目,通过实际的数据分析结果发现词条编辑微过程可以分成保守和激进两种隐藏状态,并利用提取的特征及具有变长隐状态的 Baum-Welch 算法来训练隐马尔科夫模型。利用真实词条操作历史数据集进行测试,实验结果表明基于变长隐马尔科夫模型的维基词条编辑微过程挖掘方法能够较好拟合编辑微过程,得到较好的隐马尔科夫模型推理精度。

关 键 词: 智能信息处理; 数据挖掘; 维基百科; 张量分解; 隐马尔科夫模型
中图分类号: TP311.13 **文献标志码:** A
doi: 10.16836/j.cnki.jcuit.2018.01.007

0 引言

随着互联网技术的不断进步,以人为本的 Web2.0 技术使人们在日常生活中可以自由编辑网页内容,用户不仅仅是网络信息的浏览者,更是创造者。Web2.0 技术使人们可以通过网页信息兴趣相关聚合起来,在这种模式下,用户会保持相对较高的忠诚度,无形中形成了社群[1-2]。

维基百科在默认情况下允许用户去编辑任何词条。如 2005 年 7 月 7 日,英国伦敦的交通系统发生了炸弹事故。而维基百科是最早进行报道的,一名英国的维基志愿者对此事件进行了撰写,几分钟内,其他社区志愿者就开始补充内容并且对她的错误进行修改。截止当天,超过 2500 人共同创作了一个优于任何媒体的 14 页报道。以上说明,在线协作关系可以创造出巨大的价值[3]。

目前对于 Web 的信息分析日益成熟[4-6],而对在线协作的挖掘,特别是对维基百科词条的挖掘很少见,文献[7]提出对于一个维基百科的页面,相邻的两次编辑之间的时间间隔服从一个双段幂律分布。文献[8]提出,人类的时间行为表现出非常明显的波动性和周期性。但都没有将用户的修改行为与其修改的词条发展联系起来,而文中克服了上述不足。

收稿日期: 2017-10-15
基金项目: 国家自然科学基金青年基金资助项目(61202250)

1 基于张量分解的隐马尔科夫模型隐状态挖掘

词条的每个状态都是由单个或多个用户操作组成的,用户操作可视为隐马尔科夫模型[9-11]中的观测值。同时将引入张量分解[12-13]的方法,借鉴张量分解不会破坏元素内在关系的特点构造用户操作的高阶特征体系,用交替最小二乘法[14-15](alternating least squares, ALS)对用户操作特征提取,词条的某一状态对应连续若干个非零版本,体现了操作过程中的时间特性,使用隐马尔科夫模型来构造一种能够利用状态序列的词条编辑微过程挖掘模型。

1.1 词条信息张量构建和特征提取

词条信息可以通过版本、段落、用户完成词条张量的构建,版本维度是指词条每次变换的序号,在时间维度上构成的序列; 段落维度是为便于分析和处理,从词条的整体性角度出发和考虑,以段落为单位分割; 用户维度代表了词条修改者的用户名或者用户 ip,由此可获得一个相应的稀疏张量 $X^{V\times P\times U}$,其中 V 表示版本,P 表示段落,U 表示用户。假设操作者 u 在版本 v 对第 p 段做某种操作,则 $X(v,p,u)=+$,相反地,$X(v,p,u)=\phi$,如图 1 所示。

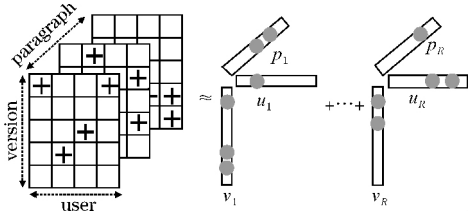


图 1 三阶词条信息张量

将三元组信息(版本、段落、用户)初始化处理构成一个三阶张量,为解决数据的稀疏性并提取特征,基于此张量的 ALS 算法具体过程如下所示。

输入: 稀疏张量 $X^{V\times P\times U}$

输出: 重构后的 $X'^{V\times P\times U}$

算法思想: 找到 R 个秩一张量来逼近 X。

算法步骤:

步骤 1 初始化因子矩阵 $\boldsymbol{U}$,$\boldsymbol{P}$;

步骤 2 重复迭代更新;

$$\hat{V} = X_{(1)}[(U\odot P)^{\mathrm{T}}]^{\dagger} \quad (1)$$

$$\hat{P} = X_{(2)}[(U\odot V)^{\mathrm{T}}]^{\dagger} \quad (2)$$

$$\hat{U} = X_{(3)}[(P\odot V)^{\mathrm{T}}]^{\dagger} \quad (3)$$

步骤 3 收敛条件检验: 若对某个误差常数 $\varepsilon > 0$,满足 $\| X - V(U\odot P)^{\mathrm{T}} \|_F^2 < \varepsilon$,则停止迭代,输出因子矩阵 U,P,V; 否则继续返回步骤 2 迭代,直至收敛。

步骤 4 计算重构张量 $X'^{V\times P\times U} = \sum_{r=1}^{R} v_{ir}p_{ir}u_{ir}$

其中 $X^{(n)}$,$n=1,2,3$ 是张量 X 的 n-模展开,符号"⊙"表示 Khatri-Rao 乘积,符号"†"表示矩阵的伪逆。

在张量降维过程中,分解操作主要为了过滤掉一些对版本进化演变过程中贡献不大的数据,而这些贡献小的数据往往是噪声,这个过程有利于提高计算的速度与精度。得到的重构 X',以此反映版本、段落、用户相互间的关联程度。

1.2　基于张量分解隐状态挖掘结果

在实验中,采集了维基百科(https://www.wikipedia.org)上"George W. Bush"词条的 2133 个历史版本,版本跨度从 2001-10-24 至 2004-5-4。原始数据集包括用户生成的内容(即操作),每个数据的信息格式整理如下: <版本号,段落,用户 ID,操作>,共计 574330 条数据。其中,move 表示仅进行移动操作,没有修改; change 表示此段内容有修改; add 表示此处新增加一段,具体信息如表 1 所示。

表 1　"George W. Bush"词条预处理数据

版本号	父版本号	段落 1	段落 2	段落 3	段落 4	段落 5
8574472	331658897	move	move	change	add	add
8574474	8574472	move	move	move	move	move
8574476	8574474	move	move	move	move	move
8574478	8574474	move	move	move	move	move
8574480	8574474	move	move	move	move	change

由于实际数据中存在大量的 move 操作,对词条本身的发展并没有很高的研究价值,利用 ALS 算法,对此张量进行降维处理。

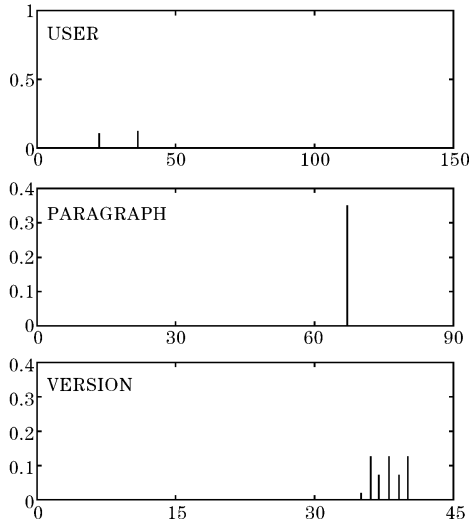


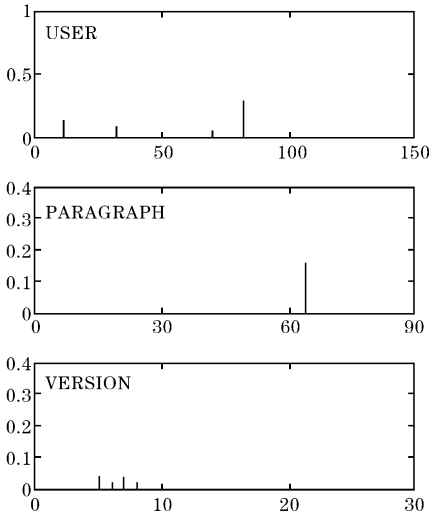
图 2　用户-段落-版本的三元关系(编辑战争)

实验结果展示了用户-段落-版本之间的三元关系,如图 2 所示。在图 2 中,两个用户之间对一段内容进行的编辑战争,以及词条编辑微过程的两种状态: 多个用户之间围绕一段内容进行的相继修改,如图 3(a); 值得注意的是第二种,多个用户在段落的维度上数量逐渐增多再逐渐变少的修改模式,如图 3(b)。该模型中的行为是由人们的自适应变化的兴趣驱动的。根据直观的分解结果,对词条的发展定为保守态和激进态。

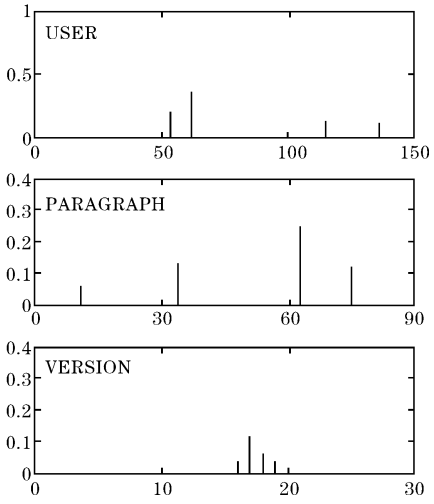
1.3　变长隐状态隐马尔科夫模型的构建

模型首先使用张量分解对数据进行去噪处理,并统计修改章节。张量分解后的用户修改行为自动聚合,再在此基础上,对具有相同修改的章节用符号"→"连接,具体词条编辑微过程推演情况如图 4 所示。可以看出词条编辑微过程主要有两种状态: 基于上一版本的修改处进行再次修改,其形状为直线式,参照图 4 部分 a,其三元关系对应图 3(a); 用户受激发进行多范围的创新操作,其形状为发散式,参照图 4 部分 b,其三元关系对应图 3(b)。

根据用户操作张量挖掘的结果,从而得到词条编辑微过程状态:



(a) 保守态



(b) 激进态

图 3 用户-段落-版本的三元关系(两种状态)

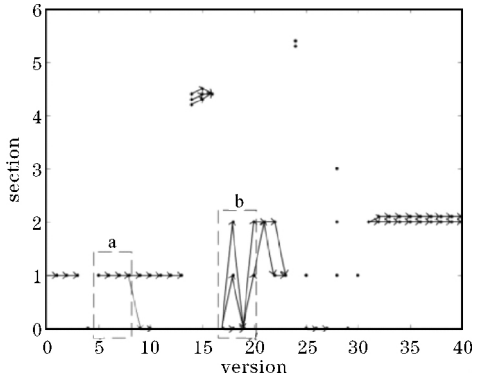


图 4 "George W. Bush"词条章节发展示意图

(1) 保守。词条处在一种平稳发展的状态,如词汇大小写、用词准确性修正等。

(2) 激进。词条处在一种活跃变化的状态,如新加标题或有创造性更新等。

定义 1 设 u 表示词条在某时刻的动态,那么所有时刻词条更新的编辑微过程状态构成隐藏状态集 $U=\{u_1,u_2\}$,其中 u_1 表示保守,u_2 表示激进。

定义 2 对于由 n 个章节组成的文档 $D=\{\mathrm{sec}_1,\mathrm{sec}_2,\cdots,\mathrm{sec}_n\}$,其中 sec_i 为第 i 章节($0<=i<=n$)。假设 Sec、Sec′为修改前后的章节集合,则有如下 5 种修改情况。

情况一 C1: $\mathrm{Sec}\cap\mathrm{Sec}'=\emptyset$

情况二 C2: $\mathrm{Sec}=\mathrm{Sec}'$

情况三 C3: $\mathrm{Sec}\cap\mathrm{Sec}'\neq\emptyset$ 且 $\mathrm{Sec}\neq\mathrm{Sec}'$

情况四 C4: $\mathrm{Sec}'\subseteq\mathrm{Sec}$

情况五 C5: $\mathrm{Sec}\subseteq\mathrm{Sec}'$

显然,所有的版本提交情况都可以归为这五种情况中的一种。设 o 表示某用户在任一时刻的修改情况,所有用户的修改操作集为 O,其中 O 的取值于集 $C=\{C_1,C_2,C_3,C_4,C_5\}$,称为观察序列。表 2 为采用定义 2 方法对不同修改版本生成的操作类型,其数据节选自图 4 中版本号为 14 至 19 的修改信息。

表 2 版本修改分类信息

版本编号	修改章节	父版本修改章节	类型
14	1	1	C_2
15	4.2,4.3,4.4	1	C_1
16	4.3,4.4,4.5	4.2,4.3,4.4	C_3
17	4.4	4.3,4.4,4.5	C_4
18	0	4.4	C_1
19	0,1,2	0	C_5

定义 3 变长隐状态隐马尔科夫用户操作模型表示为五元组(S,O,π,A,B),其中,S 称为状态序列,取值于状态集 $U=\{u_1,u_2\}$,表示词条当前发展状态;$O=\{o_1,o_2,\cdots,o_T\}$,称为观察序列,即用户的当前更新版本行为;π 为 X 的初始分布,其中 $\pi_1=P(S_1=u_i)$;A 为隐马尔科夫模型的状态转移概率矩阵,$A=(a_{ij})=P[u_j|u_i]$;每项表示词条编辑微过程状态从 $S_i(S_i\in U)$ 变为 $S_j(S_j\in U)$ 的概率;B 为观察概率矩阵,$B=(b_{i(o)})=P[o|S_i]$,每项 $b_{i(o)}$ 表示在状态 S_i 上,给定观测矢量为 o 的概率。

EM 算法是 Baum-Welch 算法循环中的一个步骤,而 Baum-Welch 是 EM 算法在隐马尔科夫模型参数学习的具体体现。在迭代过程中用张量分解对观察序列分段,将每一段的长度作为对应的隐藏状态长度,故本文使用变形 Baum-Welch 算法进行训练 $\lambda=(A,B,\pi)$,具体步骤如下所示。

输入: 用户更新行为数据 $O=\{o_1,o_2,\cdots,o_T\}$

输出: 词条编辑微过程模型参数

步骤 1 初始化模型参数 $\lambda^0=(A,B,\pi)$;

步骤 2 计算 $\xi_t(i,j)$、$\gamma_t(i)$;

$$\gamma_t(i) = P(\langle q_k\rangle_{k=t-l+1}^{t} = S_i \mid O,\lambda) \quad (4)$$

$$\xi_t(i,j) = P(\langle q_k\rangle_{k=t-l+1}^{t} = S_i, q_{t+1} = S_j \mid O,\lambda) \quad (5)$$

步骤 3 由 $\xi_t(i,j)$、$\gamma_t(i)$ 重新估计模型参数 $\lambda = (\tilde{A},\tilde{B},\tilde{\pi})$,按照公式(6)~(8)进行递推;

$$\tilde{a}_{ij} = \frac{\sum_{t}^{T-1}\xi_t(i,j)}{\sum_{t}^{T-1}\gamma_t(i)} \quad (6)$$

$$\tilde{b}_{i(o_l)} = \frac{\sum_{t=1\wedge l_o}^{T}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)} \quad (7)$$

$$\tilde{\pi}_i = \gamma_1(i) \quad (8)$$

步骤 4 反复迭代步骤 2~3,直至 $P(O\mid\lambda)$ 收敛,得到模型参数 λ。

其中 $\gamma_t(i)$ 表示 t 时刻位于词条编辑微过程状态 S_i 的概率; $\xi_t(i,j)$ 表示 t 时刻位于词条编辑微过程状态 S_i 及 t+1 时刻位于状态 S_j 的概率; l 表示 t 时刻位、于隐藏状态的长度。

2　实验设计及结果分析

首先介绍实验所用到的数据集,然后对变长隐状态的隐马尔科夫挖掘方法与传统的隐马尔科夫模型进行对比试验,并对实验结果进行了相应的分析。

2.1　测试数据集

为验证变长隐状态的隐马尔科夫的用户操作数据挖掘的有效性,采用上述张量分解所引用的实验数据。为了评价试验算法的性能,对原始数据人工标注保守或是激进。

2.2　变长隐状态 HMM 模型对比

对数据进行张量构造、分解,将分解后版本自然聚类进行分析,构造变长隐状态 HMM 模型,使用 baum-welch 算法进行训练。同时,其他 HMM 模型采用变长隐状态模型的隐状态与显状态定义进行处理数据。采用较为常用的 recall、precision 两种指标来评价模型结果的好坏。

实验的运行环境为 Intel (R) core (TM) i5-4200U CPU, 2.80GHz, 内存为 8GB, 操作系统为 Win10,使用 MATLAB R2016b、Python3.5 编程实现,结果如表 3 所示。

表 3　标准试验系统结果数据

数据	04/1/3-04/2/3		04/1/3-04/3/3		04/1/3-04/4/3		04/1/3-04/5/3	
	R	*P*	*R*	*P*	*R*	*P*	*R*	*P*
变长 HMM	0.765	0.828	0.8	0.86	0.764	0.823	0.766	0.869
HMM	0.581	0.547	0.659	0.723	0.583	0.627	0.591	0.632
2-HMM	0.671	0.645	0.592	0.601	0.656	0.674	0.688	0.705

分析时间结果可以得到:

(1) 当时间较短时,测试的版本数量较少,无法得到较为准确的结果,算法的精确率无法保证。而当数据集增加后,不同算法的召回率和精确率保持在一个范围内。

(2) 相对于 HMM 模型和 2-HMM 模型,使用张量分解后的变长隐状态隐马尔科夫模型对分析用户行为及词条编辑微过程状态识别的精确率有所提升,词条编辑微过程状态判别的错误率有所降低。

这表明固定阶数的 HMM 模型的不足之处在于将每次的修改行为独立看待,与实际人类活动的周期性与阵发性对词条编辑微过程的影响具有明显的差别,难以正确地模拟实际的词条微过程。而变长 HMM 模型对词条实际编辑微过程的模拟具有较大的优势,对这种人工标注成本较高的数据有了很好的处理。

3　结束语

采用变长隐状态的隐马尔科夫模型分析词条编辑微过程,对数据进行综合分析,将无结构的数据结构化,解决了数据的稀疏问题,并将其运用于实际词条的试验中,与固定阶数的 HMM 模型进行了对比,实验结果表明采用提出的模型在召回率和精确率比单纯使用隐马尔科夫模型更好。为分析词条编辑微过程提供了一种新思路,模型目前只适用于对词条编辑微过程趋势的简单分类,而且并未考虑时间等其他因素,这些问题都将成为下一步研究的重点。

参考文献:

[1]　WANG Wei-jun, SUN Jing. The Summarization of Research and Application of Web2.0 [J]. Informa-

tion Science,2007,12.
[2] 郑皎凌,舒红平,许源平,等.基于社群联盟的冲突消解原则求解图着色问题[J].电子科技大学学报,2016,45(1):2-16.
[3] Don Tapscott, Anthony D Williams. Wikinomics: How Mass Collaboration Changes Everything [J]. Portfolio,2006.
[4] 郑皎凌,唐常杰,姜玥,等.基于伪属性语义匹配的Deep web信息抽取[J].四川大学学报(工程科学版),2009,41(2):173-178.
[5] 郑皎凌,王鹏.Web站点核心逻辑结构挖掘[J].计算机工程,2010,36(21):57-58,61.
[6] 张朝龙,许源平,郑皎凌.基于协同过滤和文本相似性的Web文本情感极性分类算法[J].成都信息工程学院学报,2015,30(4):355-360.
[7] Zha Y,Zhou T,Zhou C.Unfolding large-scale online collaborative human dynamics [J]. Proceedings of the National Academy of Sciences of the United States of America,2016,113(51):14627.
[8] 赵飞,刘金虎,查一龙,等.在线协同写作的人类动力学分析[J].物理学报,2011,60(11).
[9] RABINER L R,JUANG B H. An introduction to hidden Markov models [J].IEEE ASSP Magazine, 1986,3(1):4-16.
[10] ZHONG A M, JIA C F.Study on the application of hidden Markov models to computer intrusion detection [A].Proceedings of the 5th World Congress on Intelligent Control and Automation [C]. Hangzhou,2004:4352-4356.
[11] 黄颖,殷瑞祥,颜刚华,等.基于GMM的与文本无关的变阈值说话人确认[J].成都信息工程学院学报,2004,(4):541-544.
[12] B W Bader,T G Kolda. Matlab tensor toolbox version2.2. Albuquerque,NM [M].USA: Sandia National Laboratories,2007.
[13] Evangelos Papalexakis,Konstantinos Pelechrinis, Christos Faloutsos.Spotting Misbehaviors in Location-based Social Networks using Tensors [J]. Companion Publication of the International Conference on World Wide Web Companion,2014: 551-552.
[14] Na Li,Stefan Kindermann,Carmeliza Navasca. Some convergence results on the Regularized Alternating Least-Squares method for tensor decomposition [J].Linear Algebra and Its Applications, 2013 (2).
[15] Kolda T G,Bader B W. Tensor Decompositions and Applications [J]. SIAM Review, 2009, 51 (3):455-500.

Wikipedia Entries Editing Micro-process Mining based on Variable Length Hidden Markov Model

HUANG Guan-ying, ZHENG Jiao-ling
(1.College of Software Engineering,Chengdu University of Information Technology,Chengdu 610225,China)

Abstract: A new method of editing microprocess of Wikipedia entries based on variablelength Hidden Markov Model was proposed. Because the traditional Expectation Maximization Algorithm needs to preset the number of hidden states, and the number of hidden states usually requires a large number of manual observations, which makes the number of hidden states set have a greater subjectivity. In this paper, we mined the number of hidden states of the micro-process by using the tensor factorization firstly. Through the actual data analysis, it is found that the editing process can be divided into two hidden states: conservative and radical. Then the extracted features and Baum-Welch algorithm with variable-length states are used to train Hidden Markov Model. The experiment results on real Wikipedia entry data show that the micro-process mining method based on the variablelength Hidden Markov Model can fit the editing micro-process well and obtain the better accuracy of the Hidden Markov Model.

Keywords: intelligent information processing; data mining; wikipedia entries; tensor factorization; hidden markov model