

基于 LDA 的软件演化确认建模

韩俊明¹ 王 炜²

(云南大学软件学院 昆明 650091)¹ (云南省软件工程重点实验室 昆明 650091)²

摘 要 演化是软件生命周期中一个重要的部分。现在有大量软件已经演化了数个版本,而如何确认演化后的软件与演化目的相符合,成为了一个需要解决的问题。由于目前还没有一个系统的方法来处理此类问题,提出了采用 LDA 主题模型的方法对演化确认进行建模分析。用 LDA 方法对软件源代码中的某些特征进行建模,通过模型能够分析出源代码内潜在的主题。将提取分析出来的主题与软件演化发布的相关报告做对比,找出它们之间的区别,以此确认演化后的软件是否符合演化目的。

关键词 软件演化,确认,LDA,主题

中图法分类号 TP311 **文献标识码** A

Method of Modeling Software Evolution Confirmation Based on LDA

HAN Jun-ming¹ WANG Wei²

(College of Software, Yunnan University, Kunming 650091, China)¹

(Key Laboratory for Software Engineering of Yunnan Province, Kunming 650091, China)²

Abstract Evolution is an important part in the software life cycle. Now, much software has evolved several versions, however, how to confirmation evolved software coincides with aim of evolution becomes a problem that calls for immediate solution. Because there is not a systematic method so far, we adopted LDA topic modeling to model analyses for evolution confirmation. LDA can model some features in the software source code, through the model the latent topics can be analyzed in the source code. We made the extracted topic compare with the published reports of software evolution to find out the distinctions between them, and according to the distinctions whether the software evolution satisfy the purpose of evolution can be confirmed.

Keywords Software evolution, Confirmation, LDA, Topic

1 引言

现代社会,软件已经成为许多行业必不可少的生产工具。但是随着社会的发展,人们的需求或者软件的运行环境会随之变化,因此软件在使用过程中不可能一成不变,这必将导致软件进行演化。软件演化由一系列复杂的变化活动组成,引起变化的原因很多:基础设施的改变、功能需求的增加、高性能算法的发现和技术环境因素的变化等^[1]。对演化后的软件,采用非人工的方法建模并通过分析该模型进行确认,这样可以节省大量的人力、物力。因此针对目前没有一种模型化方法进行确认的状况,本文提出采用 LDA 主题模型的方法,通过提取软件源代码中的一些特征,对软件演化确认进行建模。

传统的判断两个文档相似的方法一般是通过对比文档中相同单词出现的数量来实现的,但是这种判断文档相关性的方法没有考虑到文档的语义。例如潜在语义分析 LSA(latent semantic analysis)^[2] 仅仅解决了一义多词的问题,对一词多

义问题依然没有解决。因为 LSA 将每一个词表示为潜在语义空间中的一个点,因此一个词的多个意义在空间中对应的是一个点,没有被区分。

隐含狄利克雷分布 LDA(Latent Dirichlet Allocation)^[3] 是一种非监督机器学习的方法,也是主题模型,它包含了文档、主题和词 3 层结构,它可以将文档集中每篇文档的主题按照概率分布的形式给出。LDA 中每个文档被视为一系列不同主题的集合,采用忽略词间顺序的词袋方法,将文档看作为词频向量,把不能建模的文字信息转换为易于建模的数字信息。在此模型中,文档与主题间、主题与词间都构成了相应的概率分布。

LDA 主题模型采用了吉布斯采样^[4] 的方法从文档集或语料库中提取词汇。吉布斯采样适用于直接采样困难,即联合概率密度分布不可用或者很难得到的情况。吉布斯采样是一种马尔科夫蒙特卡洛(MCMC)^[5] 方法,这种方法可以接收一些观察数,这些观察数来自特定的多变量概率分布。马尔科夫链是一系列的事件集,在这个事件集中,事件一个接一个

本文受国家自然科学基金(61262024, 61379032),云南省自然科学基金(2013FB008),云南省教育厅科学研究基金(2011Y121),云南大学研究生科研课题项目(ynuy201425)资助。

韩俊明(1988—),男,硕士生,主要研究方向为软件工程、软件演化、数据挖掘, E-mail: ynwshjm@126.com; 王 炜(1979—),男,博士,副教授, CCF 会员,主要研究方向为软件工程、软件演化、数据挖掘。

的发生,并且下一个事件的发生与否仅由当前事件决定。

本文中的实验提取源代码中的特征,对特征使用 LDA 得出主题和词之间的概率关系,通过这些概率分析出潜在的主题,对比发布的相关报告,软件演化的目的就可以从代码的层面得以确认。

本文第 1 节介绍相关背景;第 2 节介绍 LDA 相关的知识;第 3 节介绍实验数据和数据的获取;第 4 节介绍实验结果并对结果进行分析;最后总结全文。

2 LDA 模型和吉布斯采样

2.1 LDA 介绍

对每一个文档集 D 中的文档 w , LDA 假设有如下生产过程^[3]:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n :
 - (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$;
 - (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n .

给定两个参数 α 和 β , 那么主题向量 θ 、主题集 N 中的一个主题 z 和词集 N 中一个词 w 的联合概率密度为:

$$p(\theta, z, w | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta) \quad (1)$$

式(1)所表达的内容可以使用 LDA 图模型表示,如图 1 所示。

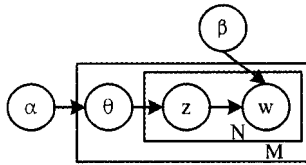


图 1 LDA 图模型

图 1 中,两个框表示重复或者迭代。外面较大的框表示文档集,里面较小的框表示在一个文档中重复选择主题和词。

对 θ 做积分,同时叠加所有的 z ,可以得到一个文档的边缘分布

$$P(w | \alpha, \beta) = \int p(\theta | \alpha) \cdot \left(\prod_{n=1}^N \sum_{z_n} P(z_n | \theta) P(w_n | z_n, \beta) \right) d\theta \quad (2)$$

对式(2)考虑所有单个文件的边缘概率分布,可以得到一个文档集的概率:

$$p(D | \alpha, \beta) = \prod_{d=1}^M \int p(\theta_d | \alpha) \cdot \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d\theta_d \quad (3)$$

式中, D 代表一个文档集或者语料库, M 代表其中文档的总数。

2.2 MCMC 方法和吉布斯采样

吉布斯采样方法是 MCMC^[6]方法的一种,也是应用最广泛的一种。吉布斯采样方法对许多复杂的贝叶斯分析问题都能直接地解决,它通过建立一条马尔科夫链,并反复迭代到该链到达均衡状态。最后获得后验分布的样本^[7]。

MCMC 的基本思想为:构造出一条马尔科夫链,它的平稳分布为待估参数的后验分布,并用该链产生后验分布的样

本,并基于该链达到平稳时的样本进行蒙特卡罗积分。设存在某一空间, n 为产生的总样本数, m 为链平稳时的样本数,则 MCMC 方法的基本思路可概括为^[8]:

1. 构造 Markov 链。构造一条 Markov 链,使其收敛到平稳分布 $\pi(x)$ 。

2. 产生样本:由其中某点出发,用步骤 1 中的链进行抽样模拟,产生点序列: $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ 。

3. 蒙特卡罗积分。对一函数 $f(x)$ 的期望估计为

$$E[f(x)] = \frac{1}{n-m} \sum_{i=m+1}^n f(x)^i。$$

对于吉布斯采样,假设 Y 为观察的样本, $\theta = (\theta_1, \dots, \theta_q)$, θ 某些分量的样本是可以没有观测到的,某些分量是参数。所需要的是 θ 或 θ 的某些分量的后验分布。吉布斯采样可以用下列步骤描述^[7]:

1. 从 $f(\theta_1 | \theta_2^{(i)}, \dots, \theta_q^{(i)}, Y)$ 中抽取一个样本 $\theta_1^{(i+1)}$;
2. 从 $f(\theta_2 | \theta_1^{(i+1)}, \theta_3^{(i)}, \dots, \theta_q^{(i)}, Y)$ 中抽取一个样本 $\theta_2^{(i+1)}$;
-

最后从 $f(\theta_q | \theta_1^{(i+1)}, \dots, \theta_{q-1}^{(i+1)}, Y)$ 中抽取一个样本 $\theta_q^{(i+1)}$ 。

若给定初值 $\theta^0 = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_q^{(0)})$, 重复上述步骤,就可以得到一个马尔科夫链。

3 实验数据获取

实验数据作为建模方法中最重要的部分,必须获取到合适的实验数据,并对其按要求进行处理。为了得到本文方法可用且有效的实验数据,或者说软件的源代码,有 3 点要求需要考虑:首先,获取到的软件源代码必须是合法的,获取方法也要合法,否则将导致实验结果的不正确;其次,该软件要有比较长的演化历史,或者有众多版本,每个版本的源代码都能得到;再次,每次更新都需要有相应的更新报告来说明本次更新的内容,以方便最后将实验结果与报告进行对比。通过查找和对比当前网络上有的开源软件,选定 jEdit 的源代码(使用 JAVA 语言编写)作为实验数据,该软件的各个版本源代码可以从其官网下载。

若直接采用软件源代码进行建模,数据量庞大,且源代码中很多都是软件功能语句或固定的代码关键字,很难得到可用率高的结果,根据文献[9]中提到的内容“源代码中的变量名、函数名、类(对象)名、API 函数、注释等关键词中蕴含了丰富的主题知识,可以通过其识别源代码与特征之间的映射关系”,本文认为源代码中能反映代码功能的特征为方法名、变量名和注释。因此需要从源代码中提取方法名、变量名和注释进行建模。

jEdit 的源代码中有 531 个类和 38 个包(采用版本 4.3,并且只从后缀名为“.java”的文件中提取特征)。考虑到某些类中只含有少量的特征信息,若将一个类作为一个文档输入并进行处理,最后的部分结果可能会由于信息过少而导致分类效果不明显,所以将一个包中的内容作为一个文档进行处理会更为适合。下列步骤为本文获取实验数据的方法:

1. 从 jEdit 的官方网站上下载源代码,其源代码格式如图 2 所示。

[illegible]

图 4 调整后的数据

4 实验和结果分析

4.1 相关实验参数

表1 LDA中的参数

org.qt.sp.jedi.bsh.JJParserState.bt	2014/5/25 23:03	文本文件	2 KB
org.qt.sp.jedi.bsh.bshThis.bt	2014/5/25 23:03	文本文件	2 KB
org.qt.sp.jedi.bsh.LHS.bt	2014/5/25 23:03	文本文件	3 KB
org.qt.sp.jedi.bsh.Modifiers.bt	2014/5/25 23:03	文本文件	1 KB
org.qt.sp.jedi.bsh.Name.bt	2014/5/25 23:03	文本文件	14 KB
org.qt.sp.jedi.bsh.NameSource.bt	2014/5/25 23:03	文本文件	3 KB
org.qt.sp.jedi.bsh.NameSpace.bt	2014/5/25 23:03	文本文件	27 KB
org.qt.sp.jedi.bsh.Node.bt	2014/5/25 23:03	文本文件	3 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	4 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	4 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	12 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	9 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	25 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	4 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	2 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	4 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	7 KB
org.qt.sp.jedi.bsh.org.objectweb.as...	2014/5/25 23:03	文本文件	8 KB
org.qt.sp.jedi.bsh.ParseException.bt	2014/5/25 23:03	文本文件	5 KB
org.qt.sp.jedi.bsh.Parser.bt	2014/5/25 23:03	文本文件	10 KB
org.qt.sp.jedi.bsh.ParserConstants.bt	2014/5/25 23:03	文本文件	2 KB
org.qt.sp.jedi.bsh.ParserTokenMana...	2014/5/25 23:03	文本文件	2 KB

```

ASN
a very small and fast Java bytecode manipulation framework
Copyright C 2000 INRIA France Telecom
Copyright C 2002 France Telecom

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation either
version 2 of the License or at your option any later version.
This library is distributed in the hope that it will be useful
but WITHOUT ANY WARRANTY without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
You should have received a copy of the GNU Lesser General Public
License along with this library; if not write to the Free Software
Foundation Inc 59 Temple Place Suite 330 Boston MA 02111307 USA
Contact Eric.Bruneau@francetelecom.com

Author Eric Bruneau

This class is a dynamically resizable vector of bytes. This class is roughly equivalent to
a DataOutputStream on top of a ByteArrayOutputStream but is more efficient.

The content of this vector

Actual number of bytes in this vector

Constructs a new link ByteVector ByteVector with a default initial size
Constructs a new link ByteVector ByteVector with the given initial size
para initialSize the initial size of the byte vector to be constructed
puts a byte into this byte vector. The byte vector is automatically
enlarged if necessary
para b a byte
return this byte vector

```

图 3 提取特征后的文件示例

4.2 实验结果

表 2 参数设置

实验输出数据将以文档的形势保存下来,具体形式和内容由表 3 所列。

表 3 输出信息

在表 3 中,“XXXXX”代表迭代次数,如果迭代次数是 0,则文件保存后的名字为“model-900. others”、“model-900. phi”、“model-900. theta”、“model-900. tassign”和“model-900. twords”。当最后一次迭代结束,“XXXXX”将被“final”所替代。

• 466 •

[1] 杨芙清,梅宏,吕建,等. 浅谈软件技术发展[J]. 电子学报,2002,30(12A):1901-1906

[2] 李阳,吴朝晖. 一种动态自适应体系结构的研究[J]. 浙江大学学报:工学版,2005,39(2):216-210,237

[3] 陈湘萍,黄翌,孙艳春,等. 支持自适应过程中分析与规划的软件体系结构模型[J]. 计算机研究与发展,2010,47:1156-1164

[4] Cheng S W. Rainbow: Architecture-based Self-adaptation with Reusable Infrastructure [J]. Computer,2004,37(10):46-54

[5] 冯耀东,黄翌,梅宏. 一种自适应软件系统结构建模及其实施方法[J]. 北京大学学报(自然科学版),2008,44(1):67-76

[6] 俞春,马骞,马晓星,等. 一种面向体系结构的软件系统自适应机制[J]. 南京大学学报(自然科学版),2006,42(2):120-129

[7] Krammer J, Magee J. Analyzing dynamic change in distributed software architectures [J]. IEEE Proceedings-Software, 1998, 145(5):146-154

[8] Garlan D, Cheng S W, Huang A C, et al. Rainbow: Architecture-based self-adaptation with reusable infrastructure [J]. IEEE Computer,2004,37(10):46-54

[9] 黄翌. 反射式软件中间件原理与技术研究[D]. 北京:北京大学,2003

[10] 马晓星. 一种面向服务的动态协同架构及其支撑平台[J]. 计算机学报,2005(4):467-477

(上接第 466 页)

图 5 示出按照表 2 的参数设置运行 LDA 程序后,所得的所有文件。文件内的信息已在表 3 中阐述。



图 5 结果文件集

表 4 列举了最后生成的文件“model-final. twords”中的部分信息,该信息为一个词以相应的概率值属于某个主题,具体可解释为:单词“search”以概率值 0.014580406400986875 属于一个特定的主题。

表 4 “model-final. twords”中部分信息

词	概率值
search	0.014580406400986875
class	0.013209745399719013
actionperformed	0.008069766644964534
actionhandler	0.006699105643696673
match	0.006185107768221225
constructor	0.006013775143062742
bufferisset	0.006013775143062742
file	0.005328444642428811
replace	0.005328444642428811

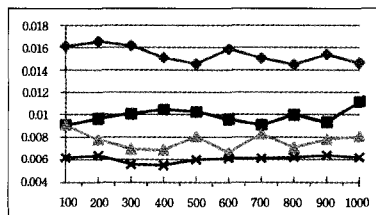


图 6 概率变化统计

图 6 是“model-final. twords”文件中的信息。从该文件中选取了 4 个词“search”(◆)、“save”(■)、“actionperformed”(▲)和“match”(×)以及在不同迭代次数下该词属于某个主题的概率,用这些数据进行绘图。图 6 中的横轴代表迭代次数,纵轴代表词属于某个主题的概率。从图 6 中可以看出,除了“save”(■)外,其他词在迭代次数较大时,其概率趋于平

稳,这说明 LDA 方法在一定的迭代次数后,会呈现明显的收敛趋势。

结束语 本文基于 LDA 主题模型的方法,提出了一种对软件演化进行确认的模型方法。首先从源代码中提取能代表软件行为的特征(注释、方法名和变量名),然后调整其格式,使用 LDA 进行处理,得出文档、主题和词的概率模型。

下载两个版本的同一软件源代码,对其进行归并处理,然后对归并后的代码利用本文方法进行建模,在产生结果的主题中,找出词或词的概率差异,即计算每个主题与各个文档的相似度,用这些计算出来的相似度与软件的更新报告中的实际被修改代码进行对比,就能从代码层面确认该软件的演化是否与用户要求或更新报告相一致。

在 LDA 主题模型中,有两个极其重要的参数 α 和 β ,其意义已在文中做了阐述。本文在使用 LDA 的过程中,根据人工的经验对这两个参数进行赋值从而失去了一般性,使得参数设置的好坏直接影响最后结果收敛与否,且针对不同的输入数据,设置的值差异较大。同时本文没有对输入数据进行词根还原、去除停词等处理,因此产生结果中会有停词和一词多式(英语中的时态,如过去式)的情况。找到一种可靠且有效的方法来解决上述两个问题,将是今后研究的重点。

参考文献

[1] 郁湧,王丽霞,赵娜. 基于构件的软件演化波及效应分析[J]. 计算机科学,2013,40(11):164-168

[2] Deerwester S, Dumais S T, Furnas G W, et al. Indexing By Latent Semantic Analysis[J]. Journal of the American Society For Information Science,1990,41(10):391-407

[3] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research,2003,3:993-1022

[4] Griffiths T L, Steyvers M. Finding scientific topics[J]. PNAS, 2004,101(1):5228-5235

[5] Brooks S P. Markov chain Monte Carlo method and its application[J]. The Statistician,1998,47(1):69-100

[6] Shimura G. 初等 Dirichlet 级数和模形式(影印版)[M]. 北京:科学出版社,2011

[7] 刘忠,茹诗松. 分组数据的 Bayes 分析-Gibbs 抽样方法[J]. 应用概率统计,1997,13(2):211-216

[8] 马海云. 基于 Gibbs 抽样的测试用例生成技术研究[J]. 自动化与仪器仪表,2011(3):11-15

[9] 何云,王伟,李彤,等. 面向行为主题的软件特征定位方法[J]. 计算机科学与探索,2014,8(12):1452-1462