第 37 卷 第 21 期
Vol.37 No.21

计 算 机 工 程
Computer Engineering

2011 年 11 月
November 2011

• 软件技术与数据库 •　　文章编号：1000—3428(2011)21—0046—03　　文献标识码：A　　中图分类号：TP311

基于 LDA 主题模型的软件缺陷分派方法

黄小亮[1]，郁抒思[1]，关佶红[2]

(1. 复旦大学计算机科学技术学院，上海 200433；2. 同济大学计算机科学与技术系，上海 201804)

摘　要：传统的基于向量空间模型的软件缺陷分派方法，由于存在特征空间维度高、数据稀疏且包含噪音等问题，分派准确率较低。为此，提出一种基于隐含狄利克雷分配(LDA)主题模型的软件缺陷分派方法，将缺陷报告从原始的高维文本单词空间映射到低维语义主题空间，在新的低维主题空间上进行分派。实验结果表明，在使用 SVM 和 KNN 分类器时，该方法的分派准确率较高。

关键词：软件缺陷分派；隐含狄利克雷分配模型；马尔可夫链蒙特卡洛方法；吉布斯采样；文本分类；向量空间模型

Software Bug Triage Method Based on LDA Topic Model

HUANG Xiao-liang[1], YU Shu-si[1], GUAN Ji-hong[2]

(1. School of Computer Science, Fudan University, Shanghai 200433, China;
2. Department of Computer Science, Tongji University, Shanghai 201804, China)

【Abstract】In traditional Vector Space Model(VSM) based software bug triage, the high dimensionality feature space are sparse and noise containing. Inspired by these characteristics, this paper proposes a software bug triage method based on Latent Dirichlet Allocation(LDA) topic model. It maps the bug report to the topic space, and makes triage in the new low dimension topic space. Experimental results show that, the method works well on bug triaging, with SVM and KNN classifiers.

【Key words】software bug triage; Latent Dirichlet Allocation(LDA) model; Markov-Chain Monte Carlo(MCMC) method; Gibbs sampling; text classification; Vector Space Model(VSM)

DOI: 10.3969/j.issn.1000-3428.2011.21.016

1　概述

大规模的开源软件，如 Eclipse 和 Firefox 等，随着规模的增大和版本的更新，会有大量的缺陷(bug)被发现和提交。由于数量很大，将这些新的缺陷分给合适的开发人员去解决，需要大量的人力和时间。软件缺陷分派的目的，就是利用缺陷跟踪系统(如 Bugzilla)中已解决缺陷的历史信息(包括参与解决缺陷的人员信息)，对新提交的缺陷进行自动分派。缺陷的自动分派能帮助系统开发与维护人员将宝贵的时间专注于缺陷的修复。

缺陷分派最常用的方法是将每一个缺陷报告看成一个文档，提取缺陷文本描述信息，然后用向量空间模型(Vector Space Model, VSM)表示软件缺陷，从而将缺陷分派转换成文本分类问题来处理。相比于普通的文本分类问题，缺陷分派可用信息少，而类别多(每个开发人员相当于一个类别)，因此分派效果普遍较差，分派准确率低。文献[1]直接将缺陷报告表示为单词的集合，使用朴素贝叶斯方法分类，取得了约 30%的准确率。缺陷分派最后的类别是开发者，可以有多个，文献[2]扩展了文献[1]的方法，根据新缺陷分到每个类的概率，取概率最高的 k 个类(开发人员)组成一个推荐列表，在 k=5 时使分派给推荐列表的准确率达到 60%左右。此外，文献[3]还使用隐含语义分析(Latent Semantic Analysis, LSA)来将缺陷报告的文本从单词空间映射到“隐含”语义空间，进行降维和去噪，然后在新的语义空间上进行分派。

为提高分派效果，本文提出一种基于隐含狄利克雷分配(Latent Dirichlet Allocation, LDA)[4]主题模型的方法，即将缺陷报告文档从高维的单词空间映射到低维的主题空间，然后进行缺陷分派。

2　软件缺陷报告与软件缺陷分派

2.1　软件缺陷报告

在大型的软件系统中都会有专门的缺陷跟踪系统，以维护整个软件系统中缺陷的基本信息和修复情况。缺陷报告是记录单个缺陷信息和状态的文本。以 Eclipse 为例，一个典型的缺陷报告中包含缺陷报告编号、出现缺陷的平台、软件版本、缺陷状态(是否修复等)、被分派给谁等信息，还有详细描述缺陷的 description 信息。同时，每个缺陷报告还有一个对应的活动日志来保存缺陷报告中信息的修改记录(如状态的改变等)。在基于文本分类的缺陷分派方法中，缺陷报告是主要信息来源。

2.2　基于向量空间模型的软件缺陷分派

使用基于文本分类的方法来进行缺陷分派时，基本方法是使用 description 信息作为文本，修复缺陷的人作为文本的类别标签，然后用 TF-IDF(Term Frequency-Inverse Document Frequency)构建向量空间模型，将每个缺陷报告表示成单词空间上的一个向量，再使用分类方法对新的缺陷报告进行分类，将其分派给类别对应的开发者。

向量空间模型利用训练集合中的所有单词来组成一个高维空间，每个不同的单词就是空间里的一个维度，每一个文档则对应空间里面的一个向量。用 description 里面的文本信息来构建向量时，先要把文本分解成一个个的单词。因为文

基金项目：国家自然科学基金资助项目(60873040)

作者简介：黄小亮(1985－)，男，硕士研究生，主研方向：数据挖掘，文本分类；郁抒思，博士研究生；关佶红，教授、博士生导师

收稿日期：2011-04-26　　**E-mail：**huangxl@fudan.edu.cn

本中有很多合成词，所以需要把这样的词分解开，判断的标准是小写字母后面跟着大写字母。同时还要剔除停用词和统一大小写，并使用提取词干的方法将不同时态的单词统一起来。停用词是指那些出现次数非常多，不具有区分意义的词，比如 of、the 等，通常由一个列表提供。

确定文档向量在各个维度上的权值时，通常采用 TF-IDF 方法，其基本思想是，某个单词在一篇文章里面出现的次数越高，同时在其他文章里面出现的次数越少，则该单词具有越好的区别能力。其中，词频指给定的文档 d 当中单词 w 出现的次数，为了防止偏向长文件，通常会除以文件总单词数。而文档频率则是指整个集合 D 中，包含 w 的文档个数。对于给定的词 w_i，它在文本 d_j 中的 *tf* 和 *idf* 值可表示为：

$$tf_{i,j}=\frac{|n_{i,j}|}{\sum_k n_{k,j}} \tag{1}$$

$$idf_i=\mathrm{lb}\frac{|D|}{|\{d:w_i\in d\}|} \tag{2}$$

其中，$n_{i,j}$ 表示单词 w_i 在文本 d_j 中出现的次数；$|D|$表示集合当中包含的总文件数；$|\{d:w_i\in d\}|$ 表示集合当中包含单词 w_i 的文件的总数，在求对数的时候，可以选用任意的底数，本文使用 2 作为底数。然后就可以得到词 w_i 的 TF-IDF 值：

$$tfi\text{-}df_{i,j}=tf_{i,j}\times idf_i \tag{3}$$

通过计算文档中每个单词的 TF-IDF 值，就可以得到文档最后的向量表示。

3　基于 LDA 主题模型的软件缺陷分派

3.1　LDA 模型

LDA 是一种对文本建模的方法，它将文档表示成一个由文档、主题和词组成的 3 层概率模型，常被用来做主题分析[5]。LDA 模型建立在文档是“词袋”(bag-of-word)的假设之上，该假设忽略了单词之间的顺序关系，是可交换的，因此，在给定某些参数的情况下，这些单词在文档中就是独立同分布的。通过 LDA 建模，可以将文本映射到主题空间上，从而对其进行主题分类和判断相似度等操作。

在构建模型时，LDA 假设 θ_d 和 φ_z 分别先验地服从参数为 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 的狄利克雷分布。狄利克雷分布是一种描述多维变量概率分布的分布，常用作概率模型中的先验假设，$\boldsymbol{\alpha},\boldsymbol{\beta}$ 是预设的参数，是一个表示多维变量相互之间权重关系的向量。如图 1 所示，主题在每个文档 d 上有一个概率分布 θ_d，单词在每个主题 z 上有概率分布 φ_z。

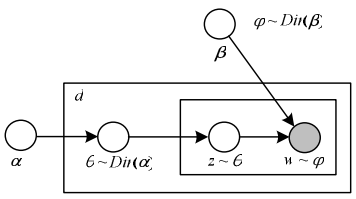

图 1　LDA 模型

在对文本构建 LDA 模型时，一种推导模型的参数的方法是使用吉布斯采样(Gibbs Sampling)的马尔可夫链蒙特卡洛(Markov-Chain Monte Carlo, MCMC)方法[6]。该方法对每个位置上的单词(将所有文档连成串)分配一个主题，并以此为状态空间来构建马尔科夫链，通过 Gibbs 采样来更新节点状态(单词的主题)，收敛到稳定状态后再用统计规律计算出数据集上 LDA 模型概率分布的近似。在构建模型时，MCMC 方法通常假设模型中的狄利克雷分布是对称狄利克雷分布，即 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 中的每个分量都取相同值，于是 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 退化为实数 α和 β。采样的具体过程为：

(1)初始化：为每个位置 i 上的单词 w_i 随机分配一个主题。

(2)更新状态：对于每一个单词 w_i，通过计算在 i 以外的其他所有单词的主题 z_{-i} 已知的情况下，w_i 属于每一个主题 j 的后验概率 $p(z_i=j\,|\,z_{-i},w)$ 来将当前单词分配给最可能的主题。

(3)迭代步骤(2)足够多次，使每个单词的主题收敛到稳定状态。

$$p(z_i\,|\,z_{-i},w)=\frac{n_{z_i}^{w}-1+\beta}{n_{z_i}-1+V\beta}\cdot\frac{n_d^{z_i}-1+\alpha}{n_d-1+T\alpha} \tag{4}$$

其中，第 1 个比值表示 w_i 属于主题 j 的比例；第 2 个比值表示文档 d 中被赋予主题 j 的单词所占的比例；T 和 V 分别表示主题数和不同单词的总数；d 是位置 i 所在的文档；$n_{z_i}^{w}$ 表示 w 分配给 z 的次数；n_{z_i} 表示所有单词分配给 z 的总次数；$n_d^{z_i}$ 为文档 d 中的单词分配给 z 的次数；n_d 是 d 包含的单词总数。经过多次采样迭代后，就能得到每个单词的主题分配情况，同时也就知道了各个文档中每个主题出现的次数。然后，就可以得出 LDA 模型中的概率分布。

$$\phi_{z,w}=\frac{n_z^{w}+\beta}{n_z+V\beta} \tag{5}$$

$$\theta_{d,z}=\frac{n_d^{z}+\alpha}{n_d+K\alpha} \tag{6}$$

其中，$\theta_{d,z}$ 就是文档 d 在主题空间模型中，主题 z 对应维度上的权重。

3.2　软件缺陷分派

基于 LDA 主题模型的缺陷分派方法利用 LDA 模型来发现当中隐含的主题信息，通过对缺陷报告建立 LDA 模型，将每个缺陷报告映射为主题空间里面的一个向量，然后在使用基于向量的分类器来对新的缺陷报告进行分派。图 2 显示了该方法的框架结构，从缺陷跟踪系统中得到缺陷报告后，先提取修复者和缺陷描述部分的信息，然后采用与 2.2 节中相同的方式进行预处理，再在上面构建 LDA 模型，得到缺陷报告在主题空间上的向量表示。

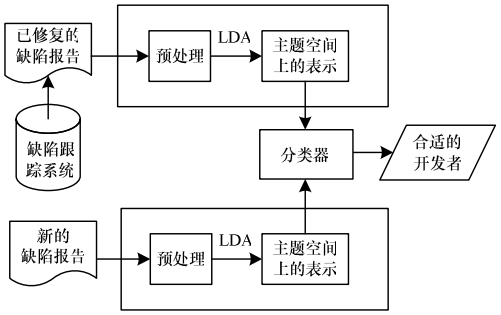

图 2　基于 LDA 模型的缺陷分派框架结构

对于一个新的缺陷报告 d^*，同样使用 Gibbs 采样的迭代方法来估计其在主题上的分布。因为训练集中的单词的主题已经稳定，所以迭代时，只需要考虑新文档 d^*里面的单词。但是在计算条件概率以更新状态的时候，需要将训练集和新文档中合并起来考虑。

$$p(z_i\,|\,z_{-i},w)^*=\frac{\tilde{n}_{z_i}^{w}-1+\beta}{\tilde{n}_{z_i}-1+V\beta}\cdot\frac{n_{d^*}^{z_i}-1+\alpha}{n_{d^*}-1+T\alpha} \tag{7}$$

其中，$\tilde{n}_{z_i}^{w}$ 和 $\tilde{n}_{z_i}$ 分别表示全部集合(训练集+新文档)中，单词

w 分配给主题 z 的次数和所有单词分配给主题 z 的总次数。然后，利用公式：

$$\theta_{d^*,z}=\frac{\tilde{n}_{d^*}^{z}+\alpha}{\tilde{n}_{d^*}+K\alpha} \tag{8}$$

就可以计算出 d^*在主题空间上的向量表示了。

4 实验及结果分析

实验选取了 Eclipse 的缺陷跟踪系统中编号 1~4 000 的缺陷报告作为样本，除去没有解决的和开发者出现次数小于 10 的部分，剩下 2 746 个样本，开发者数为 44，单词向量总维度为 5 828。使用 Gibbs 采样获取 LDA 模型参数时，α=50/T，β=200/V，T 和 V 分别表示主题数和词表长度，迭代次数为 300。原始方法用单词的 DF 值来选择特征，因为前面去除停用词时已经去掉了 DF 最高的那些词，所以这里去掉 DF 值较小而保留 DF 最大的 k 维特征，来和相同维度(30, 50, 80, 100, 150, 200, 300, 400, 500)上基于 LDA 的方法作对比。分类方法采用支持向量机(Support Vector Machine, SVM)和 K-最近邻(K-Nearest Neighbor, KNN)，取 K 为 5, 10, 20, 30, 40, 50, 60, 70 时准确率的最优值。在测试时，采用 10-folder 交叉验证，即将数据分成 10 份，每次取 9 份做训练集，剩下的为测试集，取平均准确率作为衡量结果的标准，定义为：

$$\text{准确率}=\frac{\text{分派给正确修复者的缺陷报告数}}{\text{总的缺陷报告数}}$$

因为 TF-IDF 方法在数据达到在 500 维时，使用 SVM 分类器得到的分类效果仍处于上升趋势，继续测试 700 维、1 000 维和 1 500 维的情况，得到的准确率分别为 37.54%、37.29%和 36.78%。到 1 500 维时，剔除掉的单词最大 DF 为 10，不足样本数的 0.5%，而且准确率也已经开始有所降低，因此，TF-IDF 在 SVM 上的最高准确率为 37.54%。2 种方法在不同维度上的准确率对比如图 3、图 4 所示。

实验结果表明，LDA 的效果明显好于 TF-IDF 的原始方法。在使用 SVM 分类时，TF-IDF 在 700 维上达到最好效果 37.54%，LDA 在 150 维时的准确率则为 39.45%。在低维度上，LDA 的优势更明显，维度同样为 50 时，LDA 就能够得到 38.07%的准确率，比 TF-IDF 高出 13.89%。在使用 KNN 分类时，相比于原始的 TF-IDF，LDA 方法的准确率在各个维度上也都有所提高。

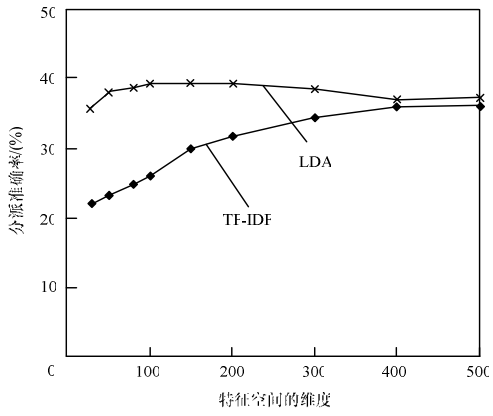

图 3 SVM 缺陷分派准确率比较

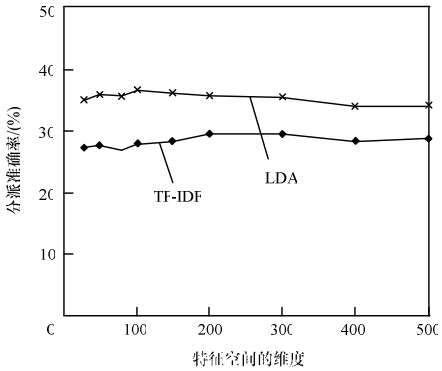

图 4 KNN 缺陷分派准确率比较

使用 LDA 将 bug 的描述映射到主题空间后，能够将同一主题下的相关词聚集到同一个维度上，这样在原来的空间上没有体现出来的相关性就得到了利用，从而克服了缺陷报告里描述文本较短、构成的向量空间过于稀疏、不利于衡量距离的缺点。观察图 3 和图 4 中的曲线可以发现，当使用 LDA 时，分派效果随着维度的增加先升后降，这是因为选择的主题太少会导致区分性不够，而主题太多以后，一些单词之间的相似关系没有得到充分利用。

5 结束语

本文提出了一种基于 LDA 主题模型的软件缺陷分派方法，将问题从原始的单词空间转换到主题空间上解决。实验结果证明，该方法能够在降低维度的同时，提高缺陷分派的准确率。在软件缺陷分派问题中，缺陷跟踪系统中的历史数据隐含了开发人员相互合作的复杂网络。在今后的工作中，将利用这些信息，结合复杂网络中的社区挖掘算法，将软件缺陷分派给相关的小组，从较粗的粒度上缩小选择范围，再在小组内进一步通过自动或人工的方式对软件缺陷进行分派，从而从整体上提高软件缺陷分派的效果。

参考文献

[1] Cubranic D, Murphy G C. Automatic Bug Triage Using Text Categorization[C]//Proc. of the 16th International Conference on Software Engineering and Knowledge Engineering. Edinburgh, UK: [s. n.], 2004.

[2] Anvik J, Hiew L, Murphy G C. Who Should Fix This Bug?[C]// Proc. of the 28th International Conf. on Software Engineering. Shanghai, China: [s. n.], 2006.

[3] Ahsan S N, Ferzund J, Wotawa F. Automatic Software Bug Triage System(BTS) Based on Latent Semantic Indexing and Support Vector Machine[C]//Proc. of the 4th International Conference on Software Engineering Advances. Porto, Portugal: [s. n.], 2009.

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

[5] 石 晶, 李万龙. 基于 LDA 模型的主题词抽取方法[J]. 计算机工程, 2010, 36(19): 81-83.

[6] Giffiths T L, Steyvers M. Finding Scientific Topics[J]. Proc. of National Academy of Science, 2004, 101(S1): 5228-5235.

编辑 顾姣健