中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

面向大规模网络数据的热点话题检测研究与系统实现

作者姓名：林尽忠

指导教师：刘玉贵 副教授

中国科学院大学计算机科学与技术学院

学位类别：工程硕士

学科专业：计算机技术

培养单位：中国科学院大学计算机科学与技术学院

2019 年 6 月

Hot Topic Detection from Large-Scale Web Data

A thesis submitted to the

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Computer Technology

in Computer Science and Technology

By

Jinzhong Lin

Supervisor: Professor Yugui Liu

School of Computer Science and Technology,

University of Chinese Academy of Sciences

June, 2019

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

随着信息技术和移动网络技术的快速发展，人们能够越来越方便地通过网络在社交媒体上获取信息和交换意，极大地促进了用户生成式内容数据的产生和传播。但是，大规模的网络数据使得用户难以从中快速有效地提取当前的热点话题以及感兴趣的话题。本文主要研究面向大规模网络数据的热点话题检测研究与系统实现，基于网络话题的无监督排序的思路，提出了两种算法来加速网络话题检测。这两种算法分别针对网络话题的快速生成和网络话题的快速排序。我们在MCG-WEBV和YKS两个数据集上验证我们算法的有效性和高效性。

针对网络话题的快速生成，我们发现网络话题在相似度空间与Lévy Walks存在统计意义上的相似性。所以，我们提出了一种无模型且无需复杂参数优化的简单算法来模拟网络话题的Lévy Walks特性。该算法基于k近邻相似度图；首先计算每个网页属于话题中心网页的概率，并按概率对网页进行排序；然后贪心地从中找出小规模的种子网页作为初始的种子话题；接着定义网页与种子话题的相似度度量方式，将每个网页分配给最相似的几个种子话题；最后按照平均相似度进行层级阈值截断来生成多粒度网络话题。通过简单的对网页进行计算和分配，我们找到了一种新的组织网络话题的方法，大大提高网络话题的生成效率，为处理大规模网络数据迈进一步。

针对网络话题的快速排序，由于泊松去卷积算法需要迭代利用所有数据来更新重构一个与相似度矩阵同等规模的矩阵，使其无法高效地处理大规模网络数据。而且由于该算法使用期望最大化算法来优化求解，导致无法使用随机优化中的随机梯度下降算法来实现可扩展性。我们发现优化最小化原则是期望最大化算法的泛化版本，所以我们将随机优化最小化算法应用到网络话题检测中，提出随机泊松去卷积算法。通过迭代地利用小批量样本来更新目标函数的代理函数，再最小化代理函数来更新求解。我们的算法减少了物理内存的需求，同时提高了计算效率，能更好地处理大规模网络数据。最后，基于多核系统，我们还实现了该算法的异步并行版本。

关键词： 网络话题，检测，可扩展性，大规模网络数据，随机泊松去卷积算法

Abstract

With the rapid development of information technology and mobile network technology, it is becoming more and more convenient for people to access information and exchange opinions on social media through the Internet, which greatly promotes the generation and dissemination of user-generated content data. However, large-scale web data makes it difficult for users to quickly and efficiently extract current hot topics and attractive topics. This paper mainly studies hot topic detection from large-scale web data. Based on the idea of unsupervised ranking of web topics, we propose two algorithms to speed up web topic detection. These two algorithms are aimed at the rapid generation and the quick sorting of web topics. We validate the effectiveness and efficiency of our algorithms on the MCG-WEBV and YKS data sets.

For the rapid generation of web topics, we find the statistically similar feature between web topics and Lévy Walks in the similarity space. Therefore, we propose a simple algorithm without model construction and complex parameter optimization to simulate the Lévy Walks feature of web topics. The algorithm is based on the k neighborhood similarity graph. Firstly, we calculate the probability that each web page belongs to the center of topic, and sort the web pages by probability； Then, we greedily find the small-scale seed web pages, which used as the initial seed topics; After that, we define the similarity measurement method of the web page and the seed topic, and assign each web page to some most similar seed topics; Finally, the hierarchical threshold is used to truncate the seed topic to generate the multi-granularity web topics according to the average similarity. By simply calculating and distributing web pages, we have found a new way to organize web topics, which greatly improve the efficiency of web topic generation and make a solid move forward in processing large-scale web data.

For the quick sorting of web topics, since the Poisson Deconvolution algorithm needs to iteratively utilize all data to update and reconstruct a matrix of the same size as the similarity matrix, it cannot efficiently process large-scale web data. Moreover, since the algorithm uses the Expectation Maximization algorithm to optimize the solution,

the Stochastic Gradient Descent algorithm in stochastic optimization cannot be used to achieve the scalability requirement. We find that the Majorization-minimization is the generalization of Expectation Maximization algorithm, so we apply the Stochastic Majorization-minimization algorithm to the web topic detection, and propose a Stochastic Poisson Deconvolution algorithm. The algorithm iteratively updates the surrogate function of objective function by utilizing small batch samples, and then minimizing the surrogate function. Our algorithm reduce the requirement of the physical memory while improving computational efficiency, and can better handling large-scale web data. Finally, based on the multi-core system, we also implemented an asynchronous parallel version of the algorithm.

Keywords: web topic, detection, scalable, large-scale web data, stochastic poisson deconvolution

目 录

图形列表

表格列表

第1章 绪论

1.1 课题研究背景

1.1.1 课题背景与意义

随着移动网络技术和社交多媒体技术的快速发展与普及，人们能够越来越方便地通过网络在社交媒体上获取信息和交换意见。比如4G移动网络的普及以及5G移动网络的推进使得网络带宽不断增加，信息传输变得更加方便快捷。同时，诸如微信、微博、抖音、快手、斗鱼等热门社交媒体APP几乎成为人手必备的软件。因此，社交媒体和移动网络技术的发展极大地促进了用户生成式数据（User-Generated Content，UGC）的产生的传播[1]。但是，海量的UGC数据使得人们难以从中快速找到自己感兴趣的内容。所以，人们希望引进某种技术来帮助人们快速、准确从海量数据中找到感兴趣的内容。例如，微博推出了热搜榜和话题榜来给用户呈现当前大多数人关注的热门话题，并取得不错的反馈。所以，以话题形式向人们呈现信息是一种有效的方式。基于此，产生了网络话题检测。网络话题检测通过对大规模的网络数据进行检测，将其中有意义的内容组织成网络话题，使得人们能够快速了解当前时事热点。

对于UGC数据，即用户生成内容数据，是指网站或其他开放性媒介的内容由其用户贡献生成。约2005年左右开始，互联网上的许多站点开始广泛使用用户生成内容的方式来提供服务。许多图片、视频、博客、论坛、社交、新闻类的网站都使用这种方式。随着互联网的发展，网络用户的交互作用得以体现。用户即是网络内容的创造者，也是网络内容的浏览者。每一个用户都可以生成自己的内容，互联网上的所有内容由用户创造，而不只是以前的某一些专业编辑。所以，互联网上的内容会飞速增长，形成一个多、广、专的局面。

对于网络话题，一般认为是针对某一现实事件的相关报道和言论的集合。围绕某一个话题的报道、言论和观点在网络上迅速传播扩散，能够在短时间、大范围内形成具有强大影响力的网络舆情。比如，针对当下热门的“奔驰女车主维权”事件是一个话题。当该事件在网络上曝光后，所有与“奔驰女车主维权”相关的信息——记者报道、用户评论、官方回应等与之相关的内容，都是该话题的组成部分。该话题在短时间被大量传播浏览，引起一片声讨，使得政

府及其有关部门立刻着手调查。同时，随着该话题不断被传播，逐渐引出一系列相关话题——“奔驰金融服务费”、“奔驰排放测试涉嫌造假”等。这大量的话题中包含了许多用户特有的行为数据，利用这些数据，可以挖掘用户的行为习惯和偏好，从而为用户构建用户画像。对于商业公司，可以利用构建的用户画像为用户推荐商品和广告；对于政府，可以用于舆情监控、敏感信息监管等途径。因此，如何快速准确地从大规模网络数据中检测热门话题是一项非常具有现实意义和研究价值的工作。

1.1.2 研究问题与难点

传统的话题检测和追踪任务（Topic Detection and Tracking，TDT）[2]致力于将每篇新闻文档分配到至少一个话题中。具体地讲，TDT是从经过专业编辑的新闻文档中生成话题[3]。而这些经过专业编辑的文档与网络数据有着极大的不同。

网络数据的特点：

(1) 规模大

任何人都可以通过网络在社交媒体上创作，且社交媒体对用户在其上发表的内容并没有太多的约束。再加上这些年快速发展的网络技术以及多媒体技术，极大地促进了网络数据的产生和传播，使得网络数据的规模越来越庞大；

(2) 约束少

网络数据由用户在社交媒体上创作产生，而社交媒体对这些创作内容并没进行格式和内容上的约束。导致网络数据受到较少的约束。

(3) 噪声多

网络数据具有很多噪声，即与主题无关的文本和图像等信息。大部分网络数据是由普通用户创作，相比专业编辑，普通用户在创作时都是很随意的。导致网络数据存在较多的噪声。

(4) 稀疏

组成网络数据的文本和视觉信息通常是简短的，在用特征表示的时候就会非常稀疏。

所以网络话题检测要处理的是大规模的网络数据，并且这些数据往往是简短的、稀疏的和充满噪声的[4]。与此同时，这大规模的网络数据中却只有很少一部分能够被组织成热点话题，大概只有5%的数据能被组织成热点话题[1]。也

就是说网络数据中还存在着大量的噪声数据。因此，网络话题检测不仅面临着大规模的数据、低效的特征，也面临着大量的噪声。这使得传统的话题检测算法[3, 5–7]不再适合处理网络数据。

一个直观的方法是在噪声存在的情况下对网络数据进行聚类。然而海量的噪声使得能够处理少量噪声的聚类算法[8–10]也不再奏效。除了网络数据外，网络话题的相应特点也会对网络话题检测带来挑战。

网络话题特点：

(1) 大小不确定

由于每个用户对于话题有不同认识，所以很难界定一个话题的大小。有的人认为强相关的数据才能构成话题，而有的人认为弱相关的数据也能构成话题。比如针对“奔驰女车主维权”的报道和“奔驰金融服务费”报道，有的人认为这是同一个话题，因为都是由奔驰车引出的一系列事件。而有的人却认为这两个报道的主要内容不一样，所以不是同个话题。因此，我们无法找到一种通用的方法来确定话题的大小；

(2) 数量不确定性

正是由于话题的多粒度性，导致不能确定一个话题的大小，从而不能确定话题的数量。而且在一个大规模的网络数据集中，我们也不可能提前明确知道话题的数量。

所以我们在解决大规模网络话题检测的时候，不仅要考虑到网络数据的特点，也要考虑到网络话题的特点。综上，针对课题研究方向，我们主要解决以下几个问题：

(1) 大规模的网络数据带来的话题检测效率低下的问题；

(2) 高噪声的网络数据带来的话题检测效果差的问题；

(3) 稀疏特征带来的低效特征问题；

(4) 话题大小不确定性问题；

(5) 话题数量不确定性问题；

1.2 常用数据集

在话题检测领域，MCG-WEBV[11]和YKS[12]是两个常用的数据集。许多网络话题检测算法使用这两个数据集来验证算法的性能。表4.1汇总了这两个数据

集的基本信息。

表 1.1 MCG和YKS数据集的基本情况汇总

数据集	话题数量	网页数量	所有话题包含网页数	词典规模	平均每个网页包含词语数量
MCG-WEBV	73	3660	832	9212	35
YKS	298	8660	990	80294	228

1.2.1 MCG-WEBV

MCG-WEBV数据集爬取了YouTube自2008年12月到2009年2月间的“浏览最多”的视频，包含了15类。同时还爬取了与这些视频相关的视频以及相同作者上传的视频。最终MCG-WEBV包含了80031个视频。

除了视频数据外，MCG-WEBV还包含丰富的信息：

- 5种元特征：视频ID、上传用户名、上传时间、视频长度、视频类别；
- 人工分类的15类视频类别及其标签；
- 8种网页特征：标题、描述、标注、评级、评论数、拍摄张数等；
- 9种视觉特征：166维颜色直方图特征、320维边缘直方图特征等；
- 采用文本特征模型产生的文本特征和36维的音频特征；

MCG-WEBV对核心数据集进行了人工标注，最终得到73个话题。这些话题由话题热度决定。而话题热度主要与话题关注度和持续时间相关。在MCG-WEBV中，话题的关注度由话题包含的视频数和视频点击数决定，话题的持续时间为话题中第一个视频的上传时间与用户最后观看时间之间的间隔。所以，话题的热度由公式1.1决定。其中$\tau(t)$指话题t的持续时间，$N(t)$表示话题t的视频总数，$V(t)$表示话题t中视频被观看的次数。

$$H(t) = \log \frac{|N(t)| * |V(t)|}{\tau(t)} \tag{1.1}$$

对于热门话题的标注，主要分为以下三个步骤：

(1) 对每个视频提取标题和标注来构建特征向量，然后通k均值聚类算法对特征向量进行无监督聚类，总共得到113个候选话题；

(2) 通过人工对这113个候选话题进行筛选，删除话题内无关的视频，对每个话题提供简单的描述；

(3) 对筛选结果进行后处理，剔除持续时间短、网页少的的话题，融合语义相近的话题，最终得到73个人工标注的网络话题。

1.2.2 YKS

YKS是一个跨媒体的多模态数据集。主要由优酷网[1]的视频数据和新浪网[2]的新闻数据共同组成。其中，约有75%的数据只包含文本信息，约有25%的数据同时包含文本信息和视觉信息。此外还有极少量的数据只包含视觉信息。

对于优酷视频，逐日爬取了从2012年5月1号开始的视频点击率在5万以上的视频，总共得到5500多个视频。然后经过过滤、剔除长度小于5秒和大于1小时的视频，最终得到2131个视频数据。除了视频外，还有其他相关的数据，比如视频标题、视频标注、视频描述、视频点击率、上传时间等。

对于新浪网的新闻数据，爬取了从2012年5月1号到2012年5月31号的新浪网发布的所有新闻。包括新闻标题、新闻正文（文本、图像、视频）和其他辅助信息，比如新闻发布时间、新闻标签、新闻点击率、新闻相关链接等。总共有30000多篇新闻文档，经过过滤空白新闻、纯图像新闻、纯视频新闻等处理后，最终剩余7325篇新闻文档。

YKS数据集同样经过话题的人工标注。在YKS数据集中，总共标注了318个话题，其中225个话题只包含新浪新闻的纯新闻，20个话题只包含优酷网的纯视频，另外73个话题同时包含新浪新闻和优酷视频。人工标注主要为一下三个步骤：对于热门话题的标注，主要分为以下三个步骤：

(1) 提取新闻或视频的文本特征，使用词袋模型表示成特征向量。然后计算特征向量之间的余弦距离；

(2) 使用k均值聚类算法对这些特征向量进行无监督聚类，总共得到400个候选话题；

(3) 对400个候选话题进行人工标注，删除话题中噪声数据，合并较小的话题以及拆分较大的话题。

经过人工标注后，过滤掉网页个数小于4的话题。最终得到318个人工标注的网络话题，同时为每个网络话题提供了简单的描述和代表性的单词。

[1]https://www.youku.com/

[2]https://www.sina.com.cn/

1.3 论文内容与组织结构

本文的整体结构如图 1.1所示，主要由以下五个部分组成：

第一章绪论：介绍了课题研究背景、常用数据集和论文内容与组织结构。其中课题研究背景包括课题研究的背景与意义以及主要解决的难点问题。常用数据集详细介绍了网络话题检测中常用的两个数据集。论文内容与组织结构介绍了本文的主要内容以及结构框架。

第二章介绍了目前话题检测领域的国内外研究现状和存在的问题。同时简要介绍了本论文给出的解决方案。

第三章针对网络话题生成部分，我们提出了新的无模型、无需复杂参数优化的算法来达到快速生成网络话题的目的。然后在两个数据集上使用多种评测标准对比我们提出的算法和当前最好的算法。

第四章针对网络话题排序部分，我们改进了原来的泊松去卷积算法，提出了一种可扩展性的随机泊松去卷积算法。该算法能够高效地处理大规模的网络数据。然后，我们构造了一个较大规模的人工数据集来验证我们算法的收敛速度。同时，我们还实现了该算法的异步并行版本，并在数据集上经验地验证其是有效的。

第五章对本文提出两个部分的工作进行总结，根据当前网络话题检测领域所出现的新特点和新问题，对下一步的研究计划进行了展望。

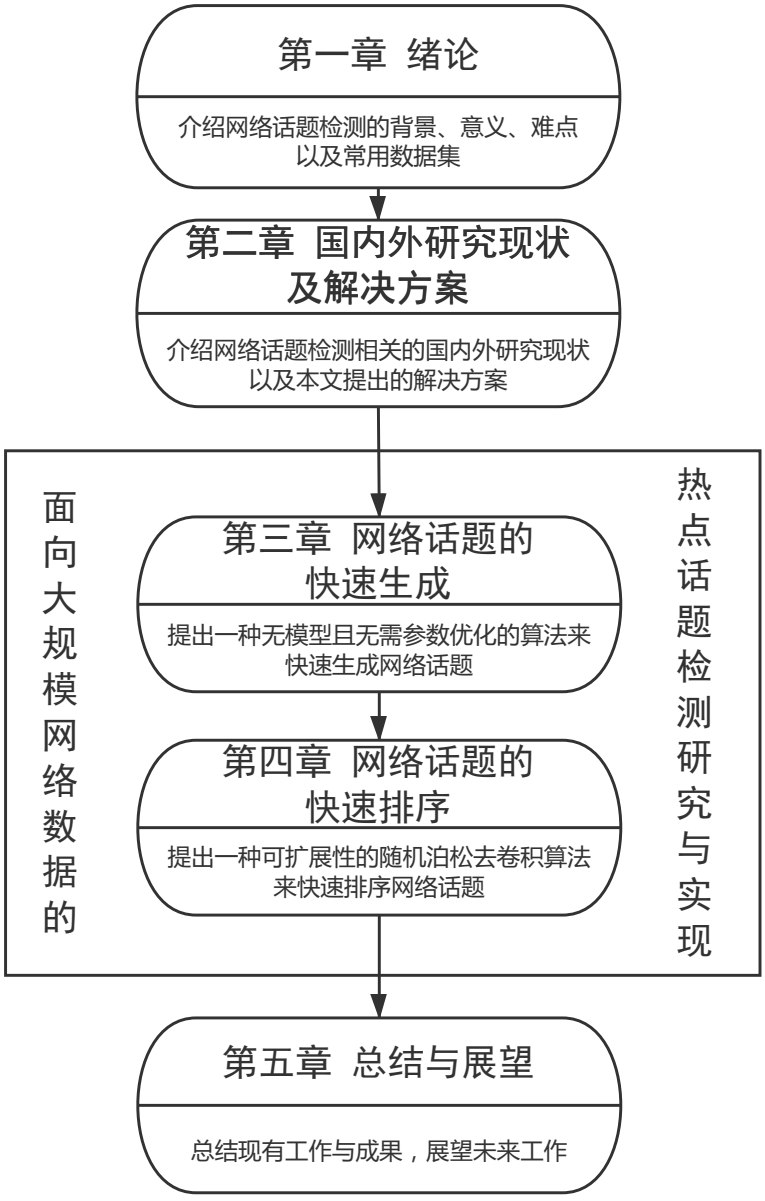

第一章 绪论
介绍网络话题检测的背景、意义、难点以及常用数据集
第二章 国内外研究现状及解决方案
介绍网络话题检测相关的国内外研究现状以及本文提出的解决方案
面向大规模网络数据的
第三章 网络话题的快速生成
提出一种无模型且无需参数优化的算法来快速生成网络话题
第四章 网络话题的快速排序
提出一种可扩展性的随机泊松去卷积算法来快速排序网络话题
热点话题检测研究与实现
第五章 总结与展望
总结现有工作与成果，展望未来工作

图 1.1 论文总体结构

第2章　国内外研究现状及解决方案

2.1　国内外研究现状

话题检测最早起源于面向事件的检测与跟踪（Event Detection and Tracking，EDT），距今已经20多年了。EDT主要检测一个特定时间发生的特定事件。然后由事件扩展演化出话题的概念，出现了话题检测与追踪任务（Topic Detection and Tracking，TDT）[2]。与EDT不同，TDT从传统对于事件的检测追踪转移到对包含突发事件及其后续报道的话题的检测与追踪。

TDT定义话题是一个核心事件或活动以及与之直接相关的事件或活动。由于当时互联网并不普及，用户更多的是通过新闻报道来了解外部世界，导致当时研究重点是自动发现新闻报道流中的话题，再按话题组织各种事件及其相应的报道。所以，TDT的主要任务就是对新闻报道进行话题检测与追踪。新闻报道大都经过专业人士创作和编辑，具有主题明确、内容正规、用词准确、噪声小、易于处理等特点。

对于纯文本数据的话题检测，主要使用自然语言处理方法，比如提取新闻文本的TF-IDF特征（Term Frequency，Inversed Document Frequency）[13]来构建特征向量，然后通过简单的聚类算法，生成的每个聚类就是一个话题。但是由于单纯的词汇统计特征并没有考虑词之间的语义关系，所以这些方法往往结果很差。考虑到文本数据中词之间的语义关联以及话题的语义性，有研究者提出了主题模型（Topic Model）。主题即我们所要检测的话题。这类模型认为文档不应该完全归属某一类，而是根据一定概率分布在隐含主题上，同时认为每个隐含主题包含多个词。其中有三个经典的主题模型：LSA、PLSA和LDA。

隐语义分析（Latent Semantic Analysis，LSA）[14]最初用在语义检索上，目的是为了找出词在文档和查询中真正的含义，也就是隐含语义，从而解决一词多义和一义多词的问题。LSA和传统的向量空间模型一样使用向量来表示词和文档，并通过向量之间的的关系来判断词之间以及文档之间的关系。不同的是，传统的向量空间模型使用精确的词匹配，即精确匹配用户输入的词与向量空间中存在的词，无法解决一词多义和一义多词的问题。因为在实际匹配中，我们想要比较的不是词，而是隐藏在词后面的意义和概念。而LSA将词和文档从高

维空间映射到低维的语义空间，再比较其相似性。从而解决一词多义和一义多词的问题，并且去除了原始向量空间中的一些噪声，提高特征的鲁棒性。实际上，如图2.1所示，LSA使用奇异值分解（Singular Value Decomposition，SVD）技术将文档-词汇矩阵*A*分解为词汇-话题矩阵*U*、话题-话题矩阵*S*、文档-话题矩阵*V*，从而挖掘出隐含的主题语义。再通过选择奇异值中最大的*t*个数，且只保留矩阵*U*和矩阵*V*的前*t*列来降维，从而达到过滤噪声和冗余数据的目的。最后基于这三个矩阵可以做语义检索、词分类和文档分类（即话题检测）。

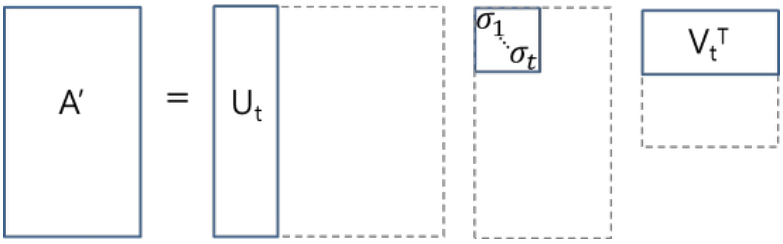


图 2.1 LSA使用SVD分解原理图

概率隐语义分析（Probability Latent Semantic Analysis，PLSA）[15]与LSA基础思想一致，都是希望找出词隐含的语义。二者区别在于LSA缺乏严谨的数理统计基础，且没有明确的物理解释，使用SVD分解操作。而PLSA使用概率模型，具有更明确的物理意义，并且使用期望最大化算法（Expectation-Maximization，EM）来学习模型参数。PLSA在文档和词之间构造隐含主题。PLSA认为一篇文档通常由多个隐含主题构成，而每个隐含主题由多个与该主题最相关的词来描述。即文档以一定的概率选择隐含主题，隐含主题以一定的概率选择词。PLSA建模思想简单，针对观察到的变量使用似然函数建模。建模中暴露出隐含变量，难以直接使用极大似然估计，所以使用EM算法求解。PLSA原理如图2.2所示。

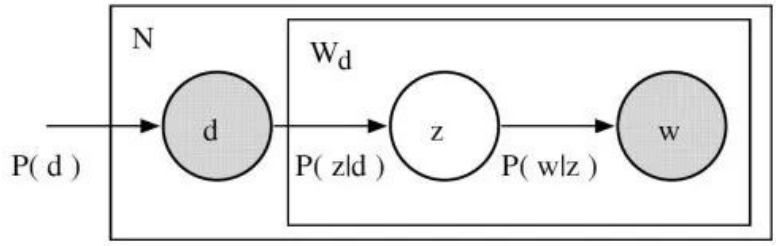


图 2.2 PLSA原理图

隐狄利克雷分布（Latent Dirichlet Allocation，LDA）[10]是PLSA的泛化版本。LDA认为文档到主题服从多项式分布，主题到词也服从多项式分布。LDA将PLSA中的参数变成随机变量，并且加入狄利克雷先验得到贝叶斯模型。使用狄利克雷先验主要是利用了狄利克雷分布和多项式分布的共轭性，方便计算。当将LDA的超参数设为特定值时，就特化成PLSA。LDA与PLSA的本质区别是估计参数的思想不同，PLSA使用频率派的思想，LDA使用贝叶斯派的思想。LDA的原理如图2.3所示。其中α和β是两个不同的狄利克雷分布的参数，分别用来生成隐含主题的分布参数θ和词的分布参数φ。z和w分别是从各自分布中选出的隐含主题和特定的单词。

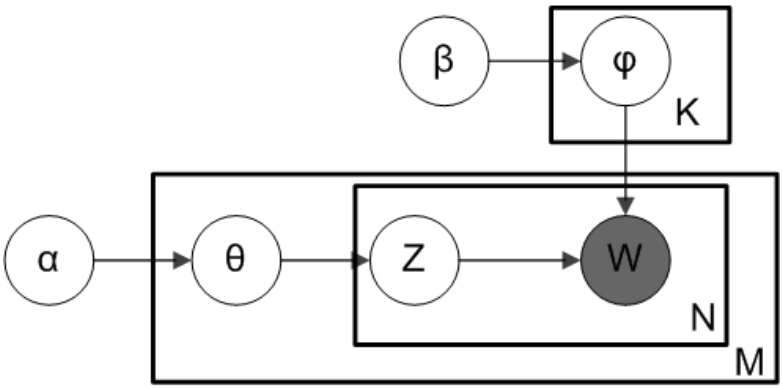


图 2.3 LDA原理图

除了上述三种主题模型外，还有层级狄利克雷过程（Hierarchical Dirichlet Processes，HDP）[16]及各种主题模型的变种。这些主题模型通常在长文本上效果较好。然而，在短文本的网络数据中的效果却非常差。这是因为短文本导致词共现较少，而这些主题模型严重依赖于词的共现性，所以主题模型不能直接用来对稀疏的网络数据进行话题检测。

同时，网络数据不再是单一模态的文本数据，更多的是诸如文本、图片、音频、视频等多模态的异构数据的融合。针对多模态数据，许多文献将网络话题检测任务当做基于多模态数据的聚类任务。有两个主流的研究路线。一个是基于多模态的方法[17, 18]，另一个是基于相似度图的方法[19]。

在基于多模态方法中，网络话题检测主要有两种研究方法：第一个研究方法是在多个模态的数据上进行聚类算法的研究。这种方法主要是将单模态的方法扩展到多模态数据。比如由LDA演变而来的多模态的LDA[18]提出从图片及其标签来检测话题。第二个研究方法是将多个模态的信息进行融合，再在融合后

的信息上进行研究[20, 21]。这种方法通过融合不同模态的信息以获得更大的信息量，再通过聚类算法检测网络话题。

在基于相似度图的方法中，多模态数据被融合进图中的边，然后将图中的顶点聚类成不同的话题。例如，Wu等人在论文[22]中通过融合来自近似重复帧（Nearly-Duplicated Keyframes，NDKs）和演讲手稿的相似度来检测新闻视频中的话题。与基于多模态的主题模型相比，基于相似度图的方法可以很容易地扩展到其他算法[19, 23, 24]。

在聚类过程时，当前比较流行的聚类定义是计算话题内部的相似度。例如，Pang等人在[1]使用相似度图中的最大团当作话题。Wang等人在[25]中使用基于成对相似度的凝聚类算法来发现新闻中的话题。Cao等人在[23]中通过k-means聚类算法在视频及其标签融合的相似度图上进行聚类。Zhang等人在[12]中提出使用图转移（Graph Shift，GS）[26]算法来寻找密集子图作为话题。这些类内相似度方法通常只能发现一小部分热点话题，导致召回率相当低。因为简单的类内相似度方法并不能很好地解决当前稀疏并且充满噪声的网络数据。

与单纯计算类内相似度相反，一些方法采用了高级的聚类算法。论文[27]使用非负矩阵分解算法（Nonnegative Matrix Factorization，NMF）来进行话题检测。与在文档上使用谱聚类算法相比，性能更好。然而这些高级方法在面对网络数据这种大规模的数据集时显得有些力不从心。无论是谱聚类算法还是非负矩阵分解算法，针对大规模数据时的复杂度非常高，导致网络话题检测效率很差。

实际上，由于网络数据的稀疏性以及高噪性，导致网络话题并不等同于聚类[23]。因此，Pang等人在论文[1]中将网络话题检测问题转化为无监督的排序问题，并提出PD算法来对话题权重进行计算。PD算法通过已有的聚类算法[28, 29]来获得过完备的话题集合，然后计算话题兴趣度，最后通过对话题的兴趣度进行排序来检测热点话题。然而，对PD算法来说，在大规模网络数据中产生过完备话题是一个非常耗时的过程[1, 30]。

尽管许多方法提出解决大规模数据的话题检测问题，但是，据我们目前所知，只有一些方法试图通过并行LDA[31–33]来解决。正如在[1, 30]所讨论的，LDA假设每个网页至少属于一个话题。然而，就网络数据而言，几乎有95%的网页不能组织成话题。因此，并行化的LDA不能去除网络数据中所包含的大量

的噪声网页。

2.2 解决方案

本论文研究面向大规模网络数据的话题检测。主要解决两方面问题，一个是针对网络话题的特点产生的问题，另一个是针对网络数据特点产生的问题。这些问题统计如下：

1 针对网络话题特点而产生的问题：

(1) 话题大小不确定性问题：主要由于每个人对话题的认识不同，导致话题大小的界定有差异；

(2) 话题数量不确定性问题：同样是由于每个人对话题的认识不同，导致话题之间的界限不明确；

2 针对网络数据特点而产生的问题：

(1) 低质量的特征表示问题：主要是由于短文本的网络数据导致稀疏的特征表示；

(2) 大量的噪声数据问题：主要是由于网络数据受到较少约束导致的大量错误、冗余、无关的数据。

(3) 大规模的数据问题：主要是由于人能够便捷地网络上随意创作而带来的网络数据的爆炸式增长。

基于上述问题，我们调研了目前国内外关于话题检测的相关文献。发现当前解决网络话题检测问题的比较优秀的方法是Pang等人在论文[1]中提出的基于无监督排序的网络话题检测算法。该算法将网络话题检测问题转换为无监督的多粒度话题排序问题，从而避免了确定话题数量和大小的问题。该算法主要分为三个阶段：构造相似度图、生成多粒度话题、通过无监督排序确定真实话题。其中在构造相似度图时只保留最相近的knn个网页，从而达到过滤噪声的目的。然后在多级相似度图上使用最大团算法（Maximum Clique, MC）生成多粒度话题。最后在相似度图上基于泊松分布实现泊松去卷积算法来计算话题权重。虽然该论文最终比其他传统方法取得更好的结果，然而没有解决低质量的特征表示问题和大规模数据问题。

随后，Pang等人在[30]中通过融合多模态信息来提高特征质量，并且采用了更高级的聚类算法——基于随机游走的非负矩阵分解算法（Nonnegative Matrix

Factorization Using Random Walk, NMFR）来生成多粒度话题。然而NMFR算法虽然能够提高聚类质量，但是同时也带来很高的时间复杂度。所以该论文还是没有解决大规模网络数据问题。

因此，本论文试图改进Pang等人提出的基于无监督排序的网络话题检测算法，以便更好地处理大规模网络数据。也就是说，本论文的最终目的是解决该算法的可扩展性问题。因此，我们主要针对该算法的两个部分进行改进。

2.2.1 生成多粒度话题

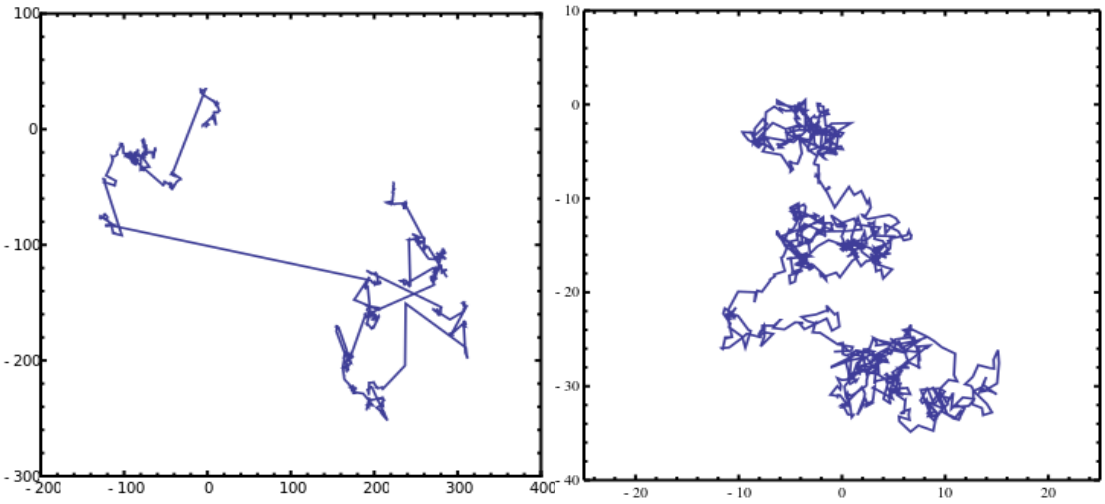
Pang等人使用过MC算法和NMFR算法在多级相似度图上生成多粒度话题。然而这两个算法都是非常耗时的算法，无法高效地处理大规模网络数据。跟传统方法一样，MC算法采用了一个看似合理的假设：相同话题内的网页之间的类内相似度应该大于话题内的网页和噪声网页之间的类间相似度。然而，由于低质量的特征表示以及语义鸿沟等问题导致这个假设站不住脚。与此同时，我们在研究真实话题在相似度空间的统计模式后，发现网络话题的组织模式与Lévy Walks存在统计意义上的相似性[34]。具体的，如果我们将网页间的相似度类比为Lévy Walks中的步长，那么一个话题内的相似度分布符合重尾分布，而重尾分布被用来定性Lévy Walks中的步长。

Lévy Walks[35, 36]是随机游走模型中的一种。其中的步长服从某种重尾的概率分布。Lévy Walks中的$flight$定义为一个质点从一个位置无偏移地移动另一个位置的步长。如图2.4展示了Lévy Walks中的$fligh$分别从柯西分布和正太分布中采样1000次的结果。直观上讲，Lévy Walks包含许多较短的$flight$和一些额外较长的$flight$。因此，Lévy Walks可以被用来描述诸如蜘蛛猴这样觅食动物的迁徙模式[35]。同时，Lévy Walks提供了一种更为有效的访问结点的方法[37]。Lévy Walks的另一个重要应用是在一个网络中路由[34]。

当Lévy Walks被用来组织网络话题时，关键问题变成：1）如何在未知分布参数情况下模拟Lévy Walks中额外较长的步长；2）如何确定所需步长的数量来组织话题；3）如何快速处理大规模的网络数据。基于上述3个问题，我们提出提出了下面对应解决方案：

(1) 我们采用了一种基于种子话题的网页多分配算法来模拟相似度空间中的Lévy Walks，优雅地避免了确定未知参数的麻烦；

(2) 我们采用了层级阈值的方法来截断话题的生长，从而产生一系列过完备



(a) 步长服从柯西分布 **(b) 步长服从正太分布**

图 2.4 1000步Lévy flight在二维坐标上的例子

的话题，保证了话题的召回率；

(3) 我们提出的网页多分配算法只需要进行简单的计算和分配，是一种无模型并且无需复杂参数优化的网络话题快速生成算法，从而向大规模网络数据的处理迈进了一步。

据我们所知，本论文是第一个发现Lévy Walks和网络话题在相似度空间上具有相似性。并且呈现了一系列完整的实验来证明这个发现对于网络话题检测的好处。我们提出的方法在计算上很简单但是效果非常好。仅仅通过简单的选择种子网页并且将网页分配到由种子网页生成的多个最相似的话题，无需进一步的参数调整等操作，我们找到一个新的组织网络话题的方式。并且在话题生成质量上比得上当前最好的方法，同时在话题生成效率上大大超过当前最好的方法。

2.2.2 无监督排序

Pang等人在论文[1]中基于泊松分布假设提出泊松去卷积算法（Poisson Deconvolution，PD）。PD算法通过扩散网页之间的相似度来对每个话题分配一个权重。虽然相似度图可以通过在线k近邻图（k-Nearest Neighborhood Graph, kN^{2}G）[38]构建，同时也可以通过稀疏矩阵的方式来高效的存储。但是，一个严重的问题是PD算法无法高效地处理大规模的网络数据。因为PD算法在每一轮必须使用所有数据在内存中重构一个$N \times N$的浮点型矩阵，其中N是网页数量。

那么我们是否可以在每一轮迭代更新时只使用一小部分数据来更新呢？一个简单但是有效的方法是随机优化[39]。这类方法至少能带来两个好处：1）减少物理内存的要求；2）避免一个$N \times N$规模的相似度图的重构。例如随机梯度下降（Stochastic Gradient Descent，SGD）及其变种[40–42]由于优秀的效果和效率，已经广泛应用于机器学习。然而，通过EM算法优化的PD算法需要保持一个和相似度图同等规模的隐变量，并且PD算法的目标函数在每轮迭代时随着期望变化而变化。所以SGD算法并不能够解决基于EM算法优化的PD算法。

优化最小化原则（Majorization Minimization，MM）[43]是EM算法的泛化版本。取目标函数的上界作为代理函数，然后迭代地最小化代理函数。许多方法可以用MM原则来解释，例如变分贝叶斯(Variational Bayes) [44]和近端算法（Proximal Algorithm）[45]。随后，论文[46]提出随机优化最小化原则（Stochastic Majorization Minimization，SMM）使得MM具有可扩展性。

受到SMM的启发，我们提出随机泊松去卷积算法（Stochastic Poisson Deconvolution，SPD）来对PD算法进行可扩展性改造。SPD迭代地更新目标函数上界构成的代理函数。最终的SPD算法不仅只需要存储一小部分采样边，而且极大地加速了算法的收敛速度。

据我们所知，本文是第一个致力于解决PD算法的可扩展性问题，并将基于代理函数的优化原则用于PD算法中。提出的SPD算法不仅概念上简单，而且非常有效。

第3章　网络话题的快速生成

3.1　引言

随着社交媒体的快速发展，越来越多的用户通过社交媒体来获取信息和分享观点。由此产生了海量的用户生成式数据[1]，使得用户难以从中快速获取热点话题及感兴趣的话题[47]。话题是某个种子事件及其相关报道的集合。网络话题检测任务[1, 12]自动地将网络数据组织成更多有意义的热门话题。从本质上来讲，网络话题检测就像从大海里捞针，类比从大量的网络数据中找到一小部分感兴趣数据并将其组织成热门事件[1]。

传统的话题检测与追踪任务[2]致力于将每个新闻报道分配到至少一个话题中[3]。而这些经过专业编辑的新闻报道数据与网络数据存在极大的差异：由于社交媒体对所发布内容的约束较少，所以来自社交媒体的网络数据更加简短、稀疏并且充满噪声[4]；在大量网页中，只有一小部分的网页能够被组织成热点话题[1]。因此网络话题检测不仅面临着低效的特征表达还需要处理大量的噪声网页。

网络话题检测的关键问题是如何在海量噪声网页存在的前提下组织热点话题。一个直观的方法是在噪声网页存在的情况下去聚类网络话题。然而海量的噪声网页使得传统方法[8, 9, 29]不再适用。为了移除大量噪声网页带来的不利影响，传统的方法[1, 12]采用了一种看似合理的假设：话题内任意一个网页与相同话题内的其他网页之间的相似度应该大于与话题外的网页之间的相似度。然而，这个假设也很难站得住脚，主要问题有以下两个：

1) 稀疏和充满噪声的网络数据导致低效的特征：用户生成式数据几乎没有受到约束，所以传统的适用于长文本的TF-IDF特征不足以高效表达社交媒体上稀疏的、充满噪声的网络数据。

2) 低阶特征和高阶语义间存在的语义鸿沟：低阶的特征难以准确地表达高阶语义间的关系。所以网页之间更大的相似度值并不意味着这两个网页在语义上更加相似。

在低效的特征表示以及海量噪声存在的前提下，我们寻找一种无模型且无需复杂参数优化的方法来生成网络话题。首先是因为网络话题的结构和内容差

异性很大，一个无模型的方法能拥有好的通用的话题生成能力；其次，为了避免高复杂度的优化措施，一个无需优化的方法能够更好地处理大规模网络数据[30]；然后，我们避免去处理短文本如何编码生成高效的特征这样一个开放性问题[48]；最后网络话题检测面临着海量的噪声。

我们研究了网络话题在相似度空间上的统计模式，发现同一个话题下的所有网页之间的相似度与Lévy Walks存在统计意义上的相似性[34]。具体地，我们将网页之间的相似度类比为Lévy Walks中的步长。那么一个热点话题中的所有相似度的分布大致服从重尾分布，而这重尾分布又是Lévy Walks中步长的特性。Lévy Walks[35, 36]是一种随机游走模型，其步长服从重尾概率分布。Lévy Walks中的$flight$定义为一个质点从一个位置无偏移的移动到另一个位置的步长。直观上讲，Lévy Walks包含许多短步长的$flight$和一些逃脱短步长控制的额外较长步长的$flight$。因此，Lévy Walks可以用来很好地描述觅食动物的迁徙模式。

当Lévy Walks被用来组织网络话题，关键问题变成如下几点：

1) 如何在未知参数下模拟Lévy Walks中额外较长的步长即较低的相似度边。

2) 如何确定话题所需要的步长个数即网页个数。

3) 在组织话题的时候如何确定所选择的这个步长是否能带来好处。

基于上述三个问题，我们提出以下解决方案：

1) 提出基于Lévy Walks的话题生成算法（Lévy Walks-based Topic Generation，LWTG）。该算法模拟Lévy Walks的特性，通过采用基于种子话题的网页多分配策略，达到除了将网页分配给最相似的种子话题以模拟Lévy Walks中普遍较短的步长，同时分配到其他几个稍微较小相似度的种子话题以模拟Lévy Walks中额外较长的步长的目的，从而优雅地避免了不同Lévy Walks需要不同参数的麻烦。简单的同时效率也很高。

2) 根据种子话题的平均相似度，使用层级阈值来截断种子话题的生长，产生一系列过完备话题，从而提高话题的召回率。

3) 定义一种度量网页跟话题相似度的标准，依据该相似度分配话题，满足一定的聚类准则。至于话题准确率是由PD算法[1, 30, 49]通过对话题的兴趣度进行排序来保证的。PD算法在经验上认为话题的精确数量不重要，一小部分热点

话题总能被正确排在前面。因此网络话题检测的准确率由PD算法保证。

据我们所知，本论文是第一个发现网络话题检测在相似度空间上和Lévy Walks具有相似的特点，并且进行了一系列实验来阐述这个发现所带来的好处。我们提出的LWTG算法不仅简单快速，而且能够进一步提高话题的召回率。通过简单地对网页进行分配，无需复杂的参数优化，我们找到了一种新的组织网络话题的方法。我们的方法在网络话题的生成效率方面已经远超当前最好的方法，而且在网络话题召回率方面也能够赶上甚至超越当前最好的方法。算法框架如图3.1所示。

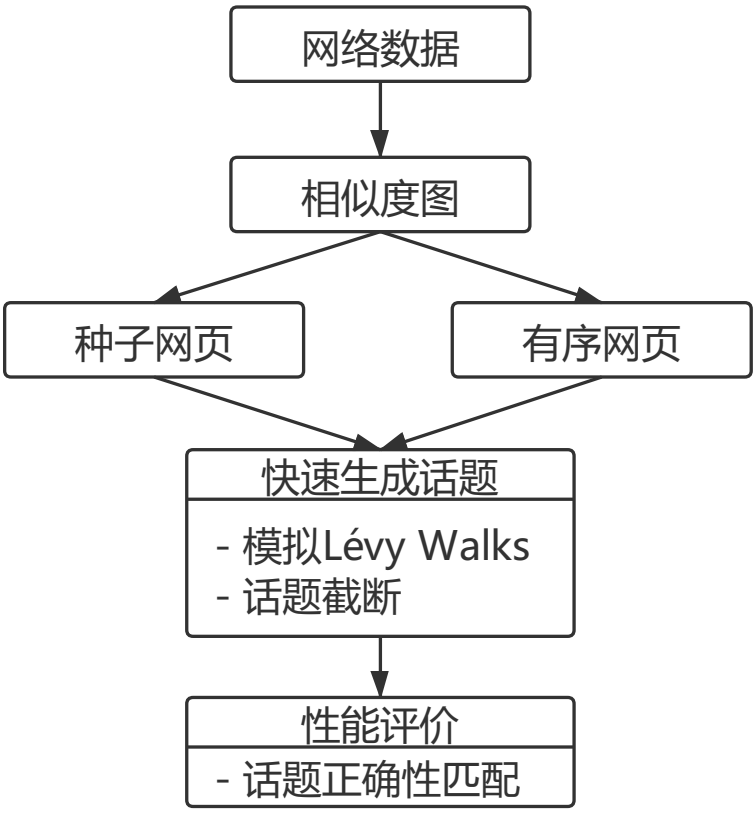


图 3.1 通过模拟Lévy Walks来快速生成网络话题的流程图

3.2 相似度图的构建

本章的研究重点是网络话题的快速生成。因此，为了减少其他因素的影响，我们忽略网络数据的多模态性和时间戳、链接等其他信息，只使用纯文本信息。所以每个网页就是一个文本字符串。对于给定的网络数据集，包含一系列的网页$W = \{w_1, ..., w_N\}$,我们对其进行处理，生成一个k近邻（k-Nearest Neighbor，k-N[2]）的相似度图$G = (V, E, A)$。其中顶点集V对应网页集合W，仿射矩阵$A(a_{ij} \in$

A)对应截断后任意两个网页之间的相似度，边集E（$e_{ij} \in E$）对应任意两个网页之间的非0边[1, 30]。具体处理如下。

首先，我们对网页的文本字符串进行分词。然后使用词袋模型[50]对网页文本进行基本的统计和表示，再用TF-IDF特征值表示每个词的权重，这样每个网页就可以用一个特征向量$x_i \in \mathbb{R}^M (i = 1, ..., N)$表示。其中M是表示词典大小，N表示网络数据集中网页的数量。对于TF-IDF特征，TF表示词频，IDF表示逆文档频率。一个词在文档中出现的频率越高，其词频值越大，相应TF-IDF值就越大。与此同时，一个词如果出现在越多的文档中，则其逆文档频率值越低，相应TF-IDF特征值就越低。逆文档频率降低那些高频出现但是较没有判别意义的词的权重。比如：‘我’、‘那’这种指示代词几乎没有判别能力，虽然它们的词频值很大，但是它们的逆文档频率值很低，进而使得TF-IDF值很低。虽然还可以使用更高级的特征，但是语义鸿沟问题仍然存在，而且使用TF-IDF特征足够简单，能够带来更快的处理速度。

然后，基于得到的每个网页的特征向量，我们可以通过余弦距离来度量两个网页之间的相似度大小。在公式3.1中，S_{ij}表示两个网页之间的初始相似度，x_i和x_j表示两个网页的特征向量：

$$S_{ij} = \frac{x_i \cdot x_j}{|x_i||x_j|} \tag{3.1}$$

最后，得到相似度图后，在每个网页中，保留与其语义最相近的k个网页关系，删除其他网页关系。这里假设两个网页之间相似度越高则语义越相近。这样做能够过滤掉大量网页之间的噪声干扰，即不相关的噪声网页。网络数据集中的噪声网页越多，k应该选择更低的值。在公式3.2中$KNN(w_i)$表示与网页w_i语义最相近的k个网页集合。同时，我们使用公式3.3将两个网页之间低于某个阈值ϵ的相似度置为0，因为我们认为过低的相似度表示这两个网页在语义上已经没有关联关系了。这进一步过滤了噪声网页。

$$S_{ij} = \begin{cases} 0, & w_j \notin KNN(w_i) \\ S_{ij}, & w_j \in KNN(w_i) \end{cases} \tag{3.2}$$

$$a_{ij} = \begin{cases} 0, & S_{ij} < \epsilon \\ S_{ij}, & S_{ij} \geq \epsilon \end{cases} \tag{3.3}$$

3.3 网络话题的Lévy Walks特性

定义 3.1. 网络话题的相似度空间

给定一个由网页数据集构建的k-N^{2}图$G = (V, E, A)$和一个话题C，网络话题C的相似度空间定义为一系列相关联的边e_{ij}经过仿射后的相似度a_{ij}，其中网页w_i和w_j至少有一个属于该话题。

网络话题的相似度空间包含两种相似度：1）话题内部相似度：同个话题内部两个网页之间的相似度；2）话题之间相似度：两个不在同个话题内的网页之间的相似度。

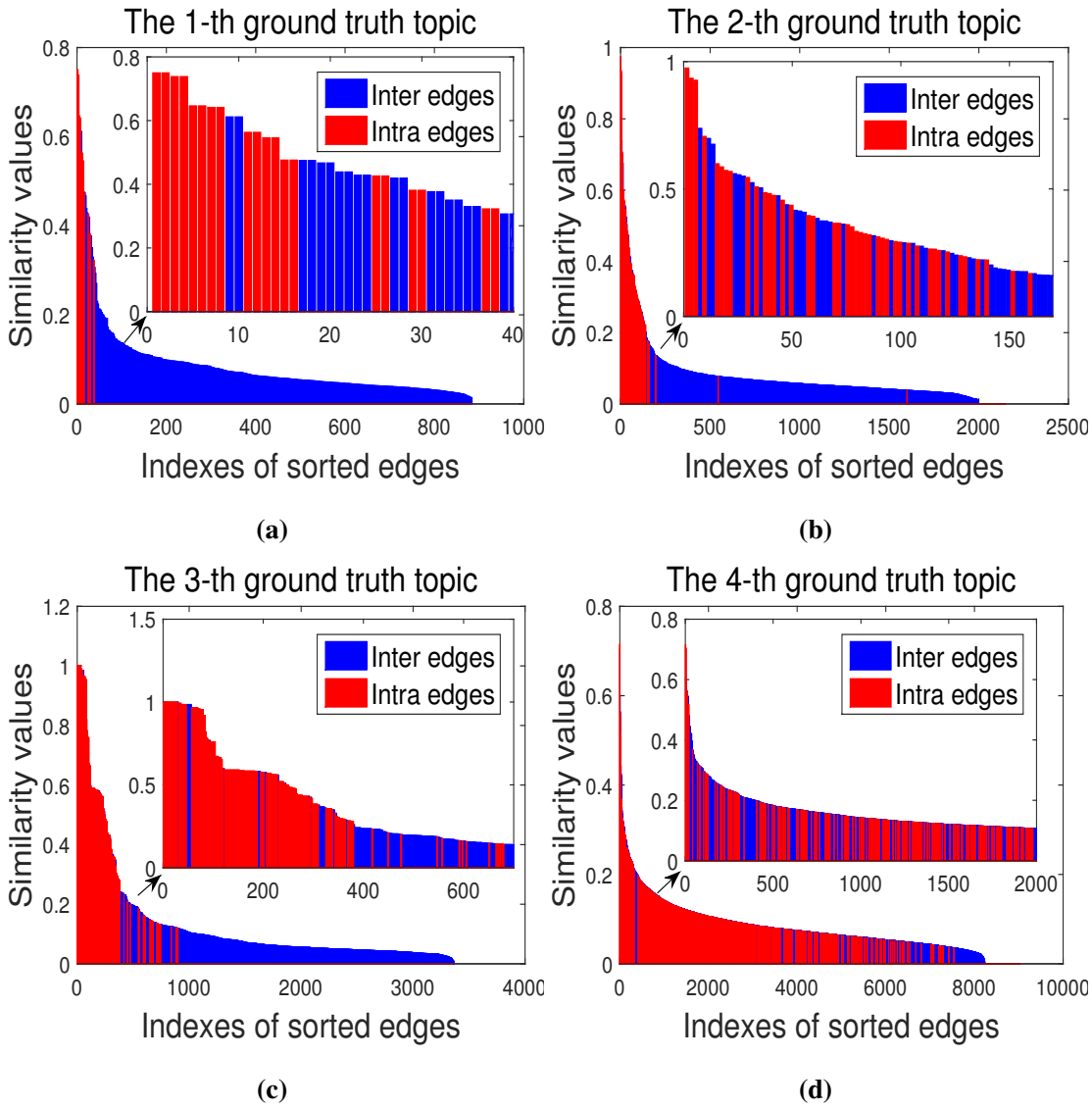
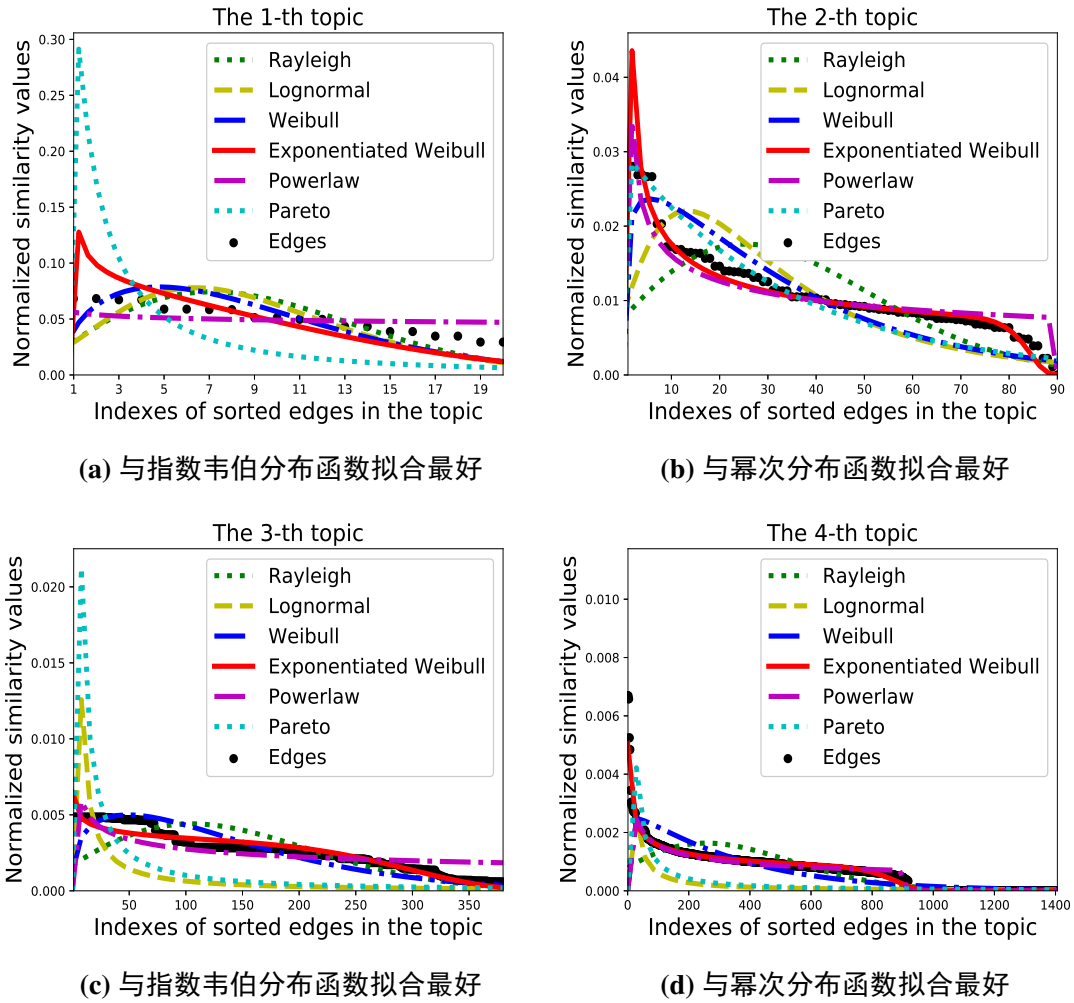


图 3.2 四个真实网络话题的相似度空间

我们从MCG-WEBV数据集中随机选择四个真实的网络话题来理解网络话题

在相似度空间上的模式。对同个话题内部的相似度进行从大到小排序。图3.2展示了这四个网络话题的相似度空间。



(a) 与指数韦伯分布函数拟合最好

(b) 与幂次分布函数拟合最好

(c) 与指数韦伯分布函数拟合最好

(d) 与幂次分布函数拟合最好

图 3.3 四个真实网络话题在相似度空间中拟合重尾分布函数

同时，对于这四个网络话题，我们将每个话题内的所有相似度归一化排序后进行分布函数拟合，从图3.3可以看出话题内已排序的相似度服从重尾分布。此时，有两个问题需要确定：

1) 是否所有的话题服从相同的重尾分布？

2) 服从相同分布的话题是否拥有相同的分布参数？

为了解决上述问题，我们使用极大似然估计去将已排序的相似度拟合为已知的分布。例如：指数韦伯分布、锐利分布、韦伯分布、对数正态分布、幂次分布、帕累托分布。表3.1列出这些分布的概率密度函数。而且，为了量化最好

的分布，我们引入赤池信息准则（Akaikeś Information Criterion，AIC）[51]：

$$\mathrm{AIC} = -2log(L(\hat{\theta}|data)) + 2K \tag{3.4}$$

其中$L(\cdot)$是似然函数，K是参数的数量。由于AIC值容易受样本大小影响[36]，不能被直接用来作为绝对的度量标准。所以，使用下面的转换形式作为每个模型的置信权重[51]：

$$w_i = \frac{exp(-\Delta_i/2)}{\sum_{r=1}^{R} exp(-\Delta_r/2)} \tag{3.5}$$

其中R是分布函数数量。$\Delta_i = \mathrm{AIC}_i - \mathrm{AIC}_{min}$，$\mathrm{AIC}_{min}$是不同AIC值中的最小值。赤池信息权重被认为是分布函数可能性的归一化值。

表 3.1 常见的重尾分布函数。

分布函数	概率密度函数
指数韦伯分布[1]	$(1-\exp(-\frac{x}{\lambda})^k)^\alpha$
锐利分布	$\frac{x}{\sigma^2}\exp(\frac{-x^2}{2\sigma^2})$
韦伯分布[2]	$\frac{k}{\lambda}(\frac{x}{\lambda})^{k-1}\exp(-(\frac{x}{\lambda})^k)$
对数正态分布	$\frac{1}{x\sigma\sqrt{2\pi}}\exp(-\frac{\ln(x)-\mu}{2\sigma^2})$
幂次分布	$a(cx)^{-k}$
帕累托分布[3]	$\frac{\alpha a^\alpha}{x^{\alpha+1}}$

[1] $\alpha \geq 1$.

[2] $k < 1$.

[3] $0 < a \leq x$

从图3.2到3.3，我们可以得到下面观察结果：

1) 相同话题下网页之间的已排序的相似度与Lévy Walks存在统计意义上的相似特性。图3.3表明同个话题内的相似度服从重尾分布。

2) 不同的话题服从不同的重尾分布。例如，根据赤池信息置信权重公式3.5，图3.3a所表示的第一个话题服从指数韦伯分布，而图3.3b所表示的第二个话题服从幂次分布。

3) 热点话题中包含一些额外的边。如图3.2所示，如果话题的相似度按照递减排序，那么由边连接排在较前的网页，并不意味其绝对属于该话题。

如果将相似度类比为网页之间的步长，那么与Lévy Walks相比，网络话题在相似度空间上有两个统计意义上的相似特性：1）Lévy Walks中的步长和网络话题的相似度均服从重尾分布；2）Lévy Walks和网络话题均包含许多较短的步长(较高相似度)和一些额外较长的步长（较低相似度）。至此，这两个相似特性被认为是网络话题的Lévy Walks特性。

3.4 通过模拟Lévy Walks生成话题

既然网络话题有Lévy Walks特性，我们试图从这个特性入手，从海量网络数据中生成话题。一个最简单的办法是根据重尾分布来组织网页进入对应话题。然而，正如之前所言，我们不可能提前训练一个通用的含有重尾分布函数的模型。

通过在相似度空间利用重尾分布的特点，我们认为可以在将网页分配给话题的时候适当添加一定的随机性以模拟Lévy Walks中额外较长的步长（较低的相似度）。为此，我们设计了一种通过模拟Lévy Walks来生成话题的算法（Lévy Walks-based Topic Generation，LWTG）。LWTG算法的框架如图3.4所示。

图 3.4 LWTG算法生成网络话题的框架图

本论文中，我们定义一种度量网页和话题相似度的方法，并将网页分配给相似度最高的K个话题。这样，除了相似度最高的话题满足聚类准则外，其他$K-1$话题用来模拟网络话题在相似度空间中的Lévy Walks特性：一些额外较低的相似度。

3.4.1 寻找种子网页

网络话题由初始核心事件不断在社交媒体上传播得以发展壮大。传播过程会逐渐吸收许多直接或者间接的外延信息。网络话题的形成可以被看做是一种信息扩散的过程。而信息扩散的过程肯定会有一定的损失。我们使用相似度流来模拟这种信息扩散过程，扩散过程中信息的损失对应于相似度流中相似度值

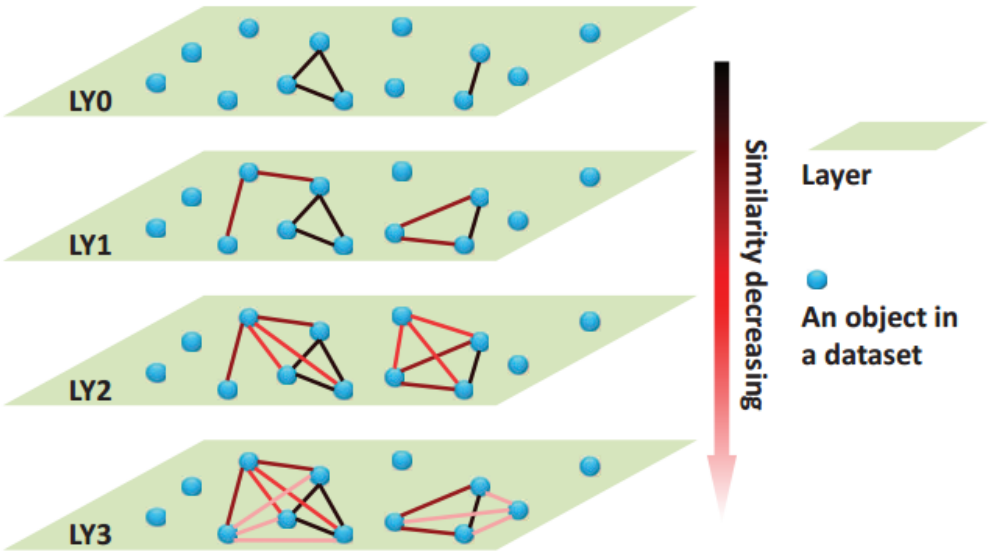


图 3.5 基于相似度流的网络话题演化示意图

的减小。图3.5展示了网络话题基于相似度流的演化过程：话题初始时只有一两个网页，随后在较低的相似度上吸收更多的网页来形成更大的话题。随着演化的进行，吸收的网页的相似度越来越低。这种过程我们称之为相似度流扩散（Similarity Cascade, SC）。又因为不同用户之间的需求是不一样的，导致用户对话题的理解具有极大的差异。所以对同一个核心事件演化形成的话题，不同的用户所理解的话题的规模也是不一样的。

既然话题是由核心事件演化来的，那么我们希望能够通过代表该核心事件的种子网页，进而通过相似度流的扩散过程来模拟演化过程，最终生成话题。首要问题是如何判断一个网页能否作为种子网页？直观上理解，在一个由多个网页构成的话题中，其网页分布应该是尽可能均匀且紧凑的[52]。均匀表示该话题内的网页之间的相似度较为接近，这是因为话题内的网页都是在为该话题服务的，所以它们应该是相似的，即相似度应该尽可能一致。紧凑表示该话题内的网页与话题应该是紧密相关，即相似度应该尽可能大。

受到论文[53]的启发，我们引进了站点熵率（Site Entropy Rate，SER）用来度量网页成为种子网页的概率。通过将相似度流从一个网页转移到另一个网页的过程模拟为全连接图中从一个站点转移到另一个站点的随机游走的过程，SER意味着从一个网页在一步内转移到其他网页的平均总信息转移量。而由种子网页吸收相似网页演化生成话题的过程中，越接近初始核心事件的网页，其

通过相似度能够转移的平均总信息量也越大。SER的公式如下：

$$\mathrm{SER}_i = \pi_i \sum_{j \in <i>} -P_{ij} \log P_{ij} \tag{3.6}$$

其中$P_{ij} = \frac{a_{ij}}{\sum_{j \in <i>} a_{ij}}$表示网页$w_i$转移到网页$w_j$的转移概率。$< i >\subset [1 : N]$保存了s个与网页$w_i$最相似的网页索引。公式3.6表明SER可以被分为两个部分：稳态分布项和熵项。这两部分作用分别如下：

1) 稳态分布项：$\pi_i = \frac{a_i}{a}$，其中$a_i = \sum_{j \in <i>} a_{ij}$是从网页$w_i$出发到s个最相关网页的相似度的和，$a = \sum_i \sum_{j \in <i>} a_{ij}$是相似度图中的所有网页及其最相似的s个网页的相似度的和。π_i被认为是网页w_i访问其他网页的频率，π_i越大，则表示由网页w_i经过一步演化的话题更加的紧凑；

2) 熵项：$\sum_{j \in <i>} -P_{ij} \log P_{ij}$度量了网页$w_i$在一步内访问其他网页的不确定性。熵项越大，表明与网页w_i直接相连的其他网页的相似度分布更加均匀。

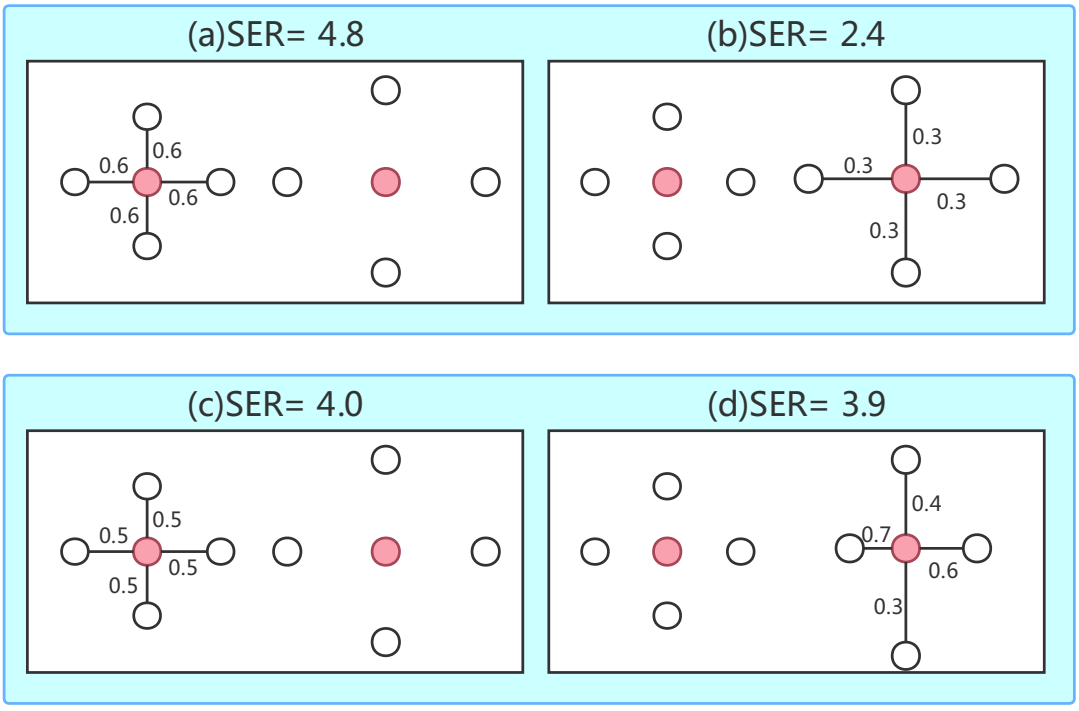


图 3.6 网页一步演化成话题的紧凑性和均匀性的SER评估

SER由这两项的乘积构成。网页的SER越大，则表示其越有可能成为话题中的中心网页。如图3.6所示，图中空心圈圈表示网页，红色实心圆圈表示要评估的网页，数字表示网页之间的相似度。从图（a）和图（b）可以看出在相同均匀分布的情况下（熵项一致），相似度分布越紧凑的网页有更大的SER值。从图

（c）和图（d）可以看出在相同紧凑性情况下（稳态分布项π一致），相似度分布越均匀的网页有更大的SER值。

定义了SER作为网页成为种子网页的概率后，我们提出了一种基于关联度的种子网页贪心搜索算法来找到不同规模的种子网页。算法1展示整个种子网页的搜索过程。算法的输入为k近邻相似度图。然后根据相似度图来计算每个网页的SER值。这里我们只需要选择与网页最相似的s个网页的相似度来计算SER值即可。因为SER仅用作一步内的相似度转移，而在一步内转移的网页应该是比较相似且数量比较少的，而不是全部的网页。所以这里我们只选择了s个最相似的网页来计算SER的值。另一个关联度参数d在搜索过程中用来确定与当前网页最相似的d个网页。只有在当前网页及其最相似的d个网页均未被标记的情况下，当前网页才被选作种子网页，同时过滤当前网页及其最相似的d个网页。这么做是因为我们认为这d个最相似的网页与当前网页同属于一个话题，是话题内部最紧密的关系，可以相互演化扩展。d值越大，就会过滤掉越多的相关网页，最终产生的种子网页数量就会越少。由于d值不好确定，同时为了更好地提高话题的召回率，我们使用多个不同的d值来产生多种规模的种子网页。算法输出为规模较小的种子网页集合。整个算法是基于SER值排序的贪心过滤算法：按SER值从大到小的顺序逐一判断每个网页，如果当前网页及其最相似的d个网页均未被过滤，则确定当前网页为种子网页，并且过滤当前网页及其最相似的d个网页。

3.4.2 网页多分配算法

在产生种子网页后，我们提出了网页多分配算法来实现种子网页吸收相似网页，进而演化成话题。我们通过将网页分配给相似度最高的K个话题来模拟网络话题Lévy Walks特性中额外较低的相似度。同时这个分配过程模拟了相似度流的扩散过程，所以我们采用了生成种子网页过程中产生的网页遍历顺序作为我们遍历网页的顺序。对于非噪声的有边连接的网页，逐个进行分配。算法2展示了这个分配过程。其中对于每个待分配的网页，我们需要计算其与种子话题的相似度，希望这个相似度能够反映种子话题对网页的吸引程度。我们定义网页w_i对种子话题C_s的相似度为该网页所带来的平均相似度对比种子话题C_s当前平均相似度的比例。如公式3.7所示，Avg(·)为取平均函数。我们认为网页w_i如果能给种子话题C_s带来所增加的相似度的平均值的比例越高，那么网页w_i与种子

算法 1 基于关联度的种子网页贪心搜索算法

Input: 相似度图 $G = (V, E, A)$，最相似网页个数s，关联度d

Output: 种子网页集合 SW，有序网页索引$SortedIndex$

初始化最终要生成的种子网页集合SW为空集
根据公式3.6计算所有网页的SER
根据SER进行从大到小排序，得到有序且有连接的网页索引$SortedIndex$
for $i \in SortedIndex$ **do**
    找出与每个网页w_i最相似的d个网页索引,记为$\{i\}$
    if 网页w_i 及其最相似的d个网页$w_{\{i\}}$ 均未被过滤 **then**
        $SW \leftarrow SW \bigcup w_i$
        标记网页w_i及其最相似的d个网页$w_{\{i\}}$
    end if
end for

算法 2 基于种子话题的网页多分配算法

Input: 相似度图$G = (V, E, A)$，网页索引$SortedIndex$，种子网页集合SW，K

Output: 一系列话题集合C

初始化空的话题集合C
初始化种子话题$C_s(s \in SW)$及对应的阈值$th_s = 1$
for $i \in SortedIndex$ **do**
    根据公式(3.7)计算S_{is} ▷ 计算网页w_i与所有种子话题的相似度
    $\{i\} \leftarrow \underset{s \in SW}{\text{argmaxk}}(S_{is}, K)$ ▷ 取与网页w_i最相似的K个种子话题的索引
    for $s \in \{i\}$ **do**
        if $IsCut(C_s, w_i, th_s)$ **then**
            $C \leftarrow C \bigcup C_s$ ▷ 截断种子话题生长，生成一个话题
            $C_s \leftarrow C_s \bigcup w_i$ ▷ 更新种子话题
            $th_s = \lfloor \text{Avg}(C_s) \times 10 \rfloor \div 10$ ▷ 更新层级阈值
        end if
    end for
end for

话题C_s必然更加密切相关。即网页w_i就有更大的可能性归属于该种子话题C_s。

$$S_{is} = \frac{\text{Avg}\left(\sum_{j \in C_s}(a_{ij} + a_{ji})\right)}{\text{Avg}(C_s)} \tag{3.7}$$

得到网页与种子话题的相似度后，我们将网页分配给相似度最高的K个种子话题。公式3.8实现了函数$IsCut(\cdot)$，其中th_s使用了层级阈值来截断种子话题，从而生成过完备的话题。阈值th_s在每一轮更新种子话题后重新赋值为当前种子话题的平均相似度所处的那一层相似度。比如说，之前种子话题的平均相似度$\text{Avg}(C_s) = 0.76$，那么该种子话题所处的相似度层的阈值为$th_s = 0.7$。如果新加的网页使得种子话题的平均相似度$\text{Avg}(C_s \cup w_i) = 0.63$，由于更新后的种子话题不再属于原相似度层（即$0.63 < 0.7$），启动阈值截断，即将之前的种子话题作为生成的完整话题加入到话题集合中去。同时，更新当前相似度层阈值为$th_s = 0.6$。

$$IsCut(\cdot) = \begin{cases} 1, & \text{Avg}(C_s \cup w_i) < th_s \\ 0, & otherwise \end{cases} \tag{3.8}$$

基于上述两个算法，我们可以得到通过模拟Lévy Walks来生成话题的算法3。其中D是种子网页关联度值的集合，用来产生多粒度的种子网页。整个基于关联度的种子网页贪心搜索算法和基于种子话题的网页多分配算法两部分组成。

算法 3 基于Lévy Walks的话题生成算法（LWTG）

Input: 相似度图$G = (V, E, A)$，最相似网页个数s，关联度集合D, 分配话题数K

Output: 过完备话题集合C

for $d \in D$ **do**

使用算法1 **/* 基于关联度的种子网页贪心搜索算法 */**

使用算法2 **/* 基于种子话题的网页多分配算法 */**

end for

3.4.3 话题排序

一旦相似度图$G(V, E, A)$构建完，我们通过算法3生成候选话题集合。然后我们在泊松噪声的假设下，使用泊松去卷积算法（Poisson Deconvolution，PD）来

评估每个话题的权重：

$$r_{ij} \sim \textbf{Poisson}(a_{ij})$$
$$s.t.: r_{ij} = \sum_{m=1}^{M} \mu_m C_{m_{ij}} \tag{3.9}$$

其中$C_{m_{ij}}$表示第m个话题是否同时包含网页w_i和w_j。话题的兴趣度由$i_m = \mu_m \cdot |C_m|$计算得到，其中C_m是第m个话题包含的网页数量。具体细节参见章节4。

3.4.4 时间复杂度分析

我们提出的算法3是基于Lévy Walks的话题生成算法。主要包含寻找种子网页和对网页进行多分配两部分。采用多粒度种子策略，其中D是种子网页的关联度集合，集合D的个数通常小于10。关联度越大，种子网页数量越少，通常从1 ~ 10中选取。

在基于关联度的种子网页贪心搜索算法中，我们需要在只保留k个近邻的相似度图中计算每个网页的SER值以及过滤相关网页，时间复杂度分别为：

- 计算SER： $O(s \cdot k \cdot N)$;
- 过滤网页： $O(d \cdot k \cdot N)$;

其中N是网页总数。k是每个网页要保留的近邻数,通常来说k小于100。s是与每个网页最相似的网页个数，通常小于20。d是要过滤的最相似网页个数（关联度），通常小于10。而对于大规模网络数据而言，网页数量N通常是巨大的。所以可以得到寻找种子网页的时间复杂度为近似线性的$O(k \cdot N)$。

在基于种子话题的网页多分配算法中，我们需要针对每个网页计算两部分内容，分别是网页与种子话题的相似度以及网页分配给种子话题，时间复杂度分别是：

- 计算网页和种子话题的相似度： $O(|topic| \cdot |SW| \cdot N)$;
- 将网页分配给k个话题： $O(K \cdot N)$;

其中N是网页总数。$|topic|$表示话题内包含的网页个数，通常小于100。$|SW|$是种子网页（种子话题）个数。 K是要分配的网页个数，通常小于5。对于大规模的网络数据而言，网页数量N通常是巨大的。所以可以得到网页多分配算法的时间复杂度为$O(|SW| \cdot N)$

从上面时间复杂度分析可以看出在算法3中，基于种子话题的网页多分配算

法占据主要的时间复杂度。因此我们提出的LWTG算法的时间复杂度为$O(|SW| \cdot N)$。种子网页的个数SW小于网页数N的一半。

3.5 实验验证

本节对我们提出LWTG算法在MCG-WEBV和YKS这两个数据集上展开实验。主要进行两方面的比较。第一个是跟两个优秀的能够处理轻量噪声数据的聚类算法进行对比；第二个是跟其他三个优秀的网络话题检测算法进行对比。通过这两类对比实验来验证我们算法的性能。

3.5.1 数据集预处理

对于MCG-WEBV数据集，我们使用其中的文本数据包括标题、标签和描述。首先过滤掉文本数据中的停用词，由于该数据集基本由英文构成，所以使用Python中的NLTK模块，对每个单词提取词干、统计TF-IDF值作为单词权重，生成每个网页的特征向量。

对于YKS数据集，基本由中文组成。所以需要采用NLPIR系统对文本数据进行预处理，包括分词、去停用词、处理同义词和扩展词等，然后统计每个词的TF-IDF的值，生成每个网页的特征向量。

基于生成的特征向量，按照3.2生成k近邻相似度图。

3.5.2 评测标准

在评测标准上，我们使用以下两种评测指标：

• 最高10个检测话题的F_1分数的均值-检测的话题数量（Top10-F_1 v.s. Number of Detected Topics，NDT）：对于每个检测得到的话题D_t，对应其最高匹配程度的真实标注的话题G_t，我们可以得到每个检测到的话题的F_1分数的公式3.10：

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.10}$$

其中精确率$Precision$定义为公式3.11：

$$Precision = \frac{|D_t| \bigcap |G_t|}{|D_t|} \tag{3.11}$$

召回率$Recall$定义为公式3.12：

$$Recall = \frac{|D_t| \bigcap |G_t|}{|D_t| \bigcup |G_t|} \tag{3.12}$$

其中$|\cdot|$表示一个话题中的网页数目。

- 准确率-平均到每个话题上的误检率（*Accuracy* v.s. False Positives Per Topics，*FPPT*）：准确率*Accuracy*的公式定义为3.13：

$$Accuracy = \frac{\#\text{Successful}}{\#\text{Groundtruth}} \tag{3.13}$$

*FPPT*的公式定义为3.14:

$$FPPT = \frac{\#\text{Detected} - \#\text{Successful}}{\#\text{Successful}} \tag{3.14}$$

其中话题被认为是正确检测到的标准由*NIR*（Normalized Intersected Ratio）指标衡量。*NIR*指标定义为公式3.15：

$$NIR = \frac{|D_t| \bigcap |G_t|}{|D_t| \bigcup |G_t|} \tag{3.15}$$

当检测到的话题的*NIR*高于一定阈值（通常设为0.5）[1]时，我们认为这是一个正确检测到的话题。对于#△表示对应集合△的元素数量。

对于这两种评测指标，Top10-F_1 v.s. *NDT*衡量算法检测到的最好的前10个话题的性能，并没有考虑到检测过程带来的误检率。而*Accuracy* v.s. *FPPT*综合衡量了所检测到的话题的召回率以及相应的平均每找到一个正确话题所带来的误检数。在这两种指标中，当具有相同Top10-F_1分数或*Accuracy*值的时候，更低*NDT*或*FPPT*的算法具有更优的话题检测效果。

3.5.3 实验设置

在实验中，我们选择了网页的文本数据进行词汇的TF-IDF统计和编码，然后使用余弦距离构建相似度图。最后对每个网页只保留最相似的k个网页的相似度值构建一个近邻图。这里对MCG-WEBV数据集的k设为100，最相似网页个数s设为10。对YKS数据集，由于噪声相比MCG-WEBV数据集更严重，所以近邻值k设为15，最相似网页个数s设为15。在所有实验中，同时在k近邻相似度图上过滤噪声网页的阈值ϵ设为0.1。跟种子粒度相关的关联度参数集合D设为$\{1, 2, 3, 4\}$。网页多分配的话题数K为2。

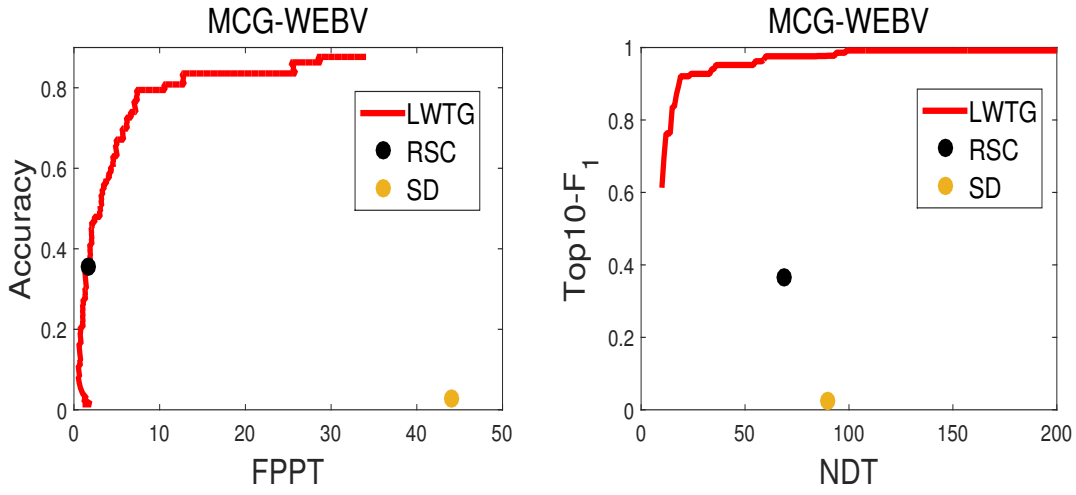
3.5.4 与聚类算法的对比

我们对比了LWTG算法与两个性能优秀的能够处理轻量噪声数据的聚类算法：

a) **Robust Spectral Clustering (RSC) for noisy data[8]**。这篇论文通过对谱聚类中的相似度图进行稀疏和隐式分解来处理噪声。然而，这种方法假设了噪声是稀疏的，但是在网络话题检测的场景下，大概95%的网页都是噪声网页。

b) **Skinny-Dip (SD)[9]**。SD基于检验分布是否为单峰分布的Hartigan's elegant dip test，从而得到一个有效的特征集来聚类。

注意到RSC和SD算法不是为了检测网络话题而是为了从噪声数据中进行聚类。所以对于RSC，SD和LWTG的对比，主要是为了验证能处理噪声数据的聚类算法，能否有效地在海量噪声数据中检测到网络话题。在下面的实验中，对RSC，聚类个数设为真实话题个数。如MCG-WEBV数据集的73个真实话题和YKS数据集的298个真实话题。对于SD，聚类个数由算法自动确定。

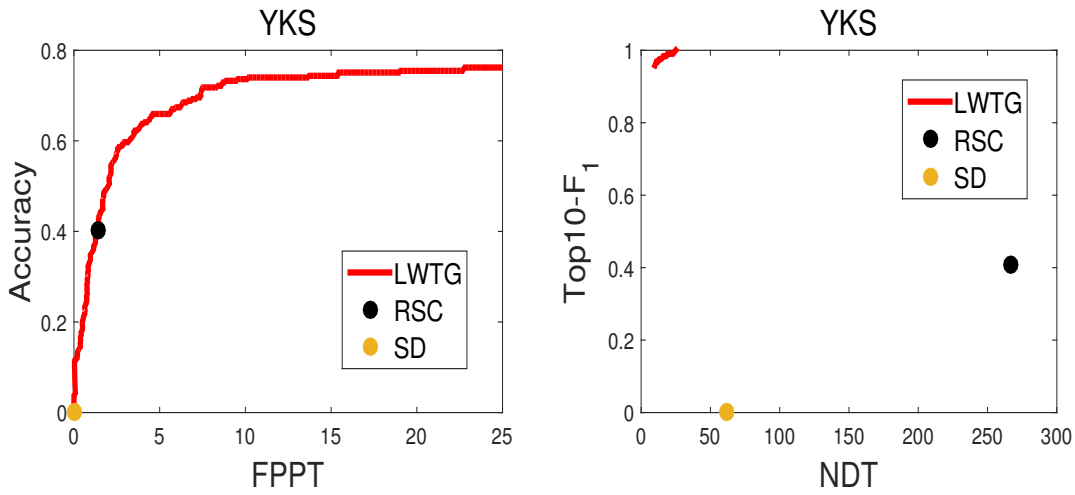


(a) **(b)**

图 3.7 LWTG算法和聚类算法RSC、SD在MCG-WEBV数据集上的对比

图3.7a和图3.7b展示了LWTG算法和RSC、SD算法在MCG-WEBV数据集上的实验结果，表明我们的LWTG算法优于SD和RSC算法。从图中可以看出SD算法的*Accuracy*和Top10-F_1几乎为0。这是因为在MCG-WEBV数据集中，TF-IDF特征的维度是9,212，这使得SD算法难以有效地从高维和充满噪声的特征中发现聚类。同时，注意到RSC算法在*Accuracy*上跟我们的算法结果相差不大。但是，在Top10-F_1上却差很多。这是因为由RSC算法生成的聚类的质量受到大量噪声的影响。而我们的方法不需要复杂的模型构建和优化就能在这两种评测标准上都能取得更好的结果。

图3.8a和图3.8b进一步在YKS数据集上对比LWTG算法和RSC、SD算法。注



(a) **(b)**

图 3.8 LWTG算法和聚类算法RSC、SD在YKS数据集上的对比

意到SD算法的实验结果纵轴为 0 ，这是因为YKS数据集的特征维度高达80,294，使得SD方法无法找到任何有效聚类。图3.8a和图3.8b表明我们的算法在YKS数据集上仍然比RSC和SD算法好。

总之，从话题检测质量来看，在相同话题数量的情况下，我们的LWTG算法在MCG-WEBV和YKS两个数据集上均比RSC和SD方法取得更好的结果。

表3.2进一步在3.6Hz CPU和32G内存的电脑上对比了RSC、SD和LWTG算法的运行时间。虽然在图3.7和图3.8中，当FPPT约等于3时，RSC算法的话题准确率和我们的LWTG算法差不多，但是我们的LWTG算法的最终话题召回率远远高于RSC算法。例如表3.2显示我们的LWTG算法在MCG-WEBV数据集上的话题召回率能够高达0.88，而RSC却只能达到0.35。这是因为我们的算法通过多种规模的种子网页来模拟网络话题的Lévy Walks特性，从而保证了话题检测系统的召回率。

我们同时还注意到当网页从MCG-WEBV数据集的3,600增长到YKS数据集的8,660的时候，RSC算法花了超过15（如：125/8）倍的时间。作为对比，我们的LWTG算法使用了大概5（如：333/73）倍的时间。

总之，从话题生成效率来看，虽然我们的LWTG算法比RSC算法慢。但是，当网页数量不断增加时，RSC算法比LWTG算法更加敏感。

表 3.2 不同话题生成算法的运行时间（秒）和系统准确率（*Accuracy*）对比

Data set(#webpages)	LWTG(*Accuracy*)	RSC(*Accuracy*)	SD(*Accuracy*)
MCG-WEBV(3660)	73(**0.88**)	**8**(0.35)	20(0.02)
YKS(8660)	333(**0.77**)	**125**(0.40)	152(0.00)

3.5.5 与网络话题检测算法的对比

我们对比了LWTG算法与3个优秀的网络话题检测算法：

a) **Multi-Modality Graph (MMG)[54]。** 该算法属于多模态网络话题检测。Zhang等人首先利用视频的NDK信息和文本信息建立相似度图，然后使用图转移算法（Graph Shift，GS）[26]进行话题检测。MMG假定了一个话题内的元素应该密切相关。所以，通常情况下MMG产生的话题规模较小。

b) **PD with Non-negative Matrix Factorization on Graph (NMFG)[49]。** NMFG和LWTG最大的不同的是产生话题的方式。NMFG中使用基于图的非负矩阵分解（Non-negative Matrix factorization on Graph，NMG[28]）在相似度级联上生成过完备话题。NMG算法的聚类非常耗时。

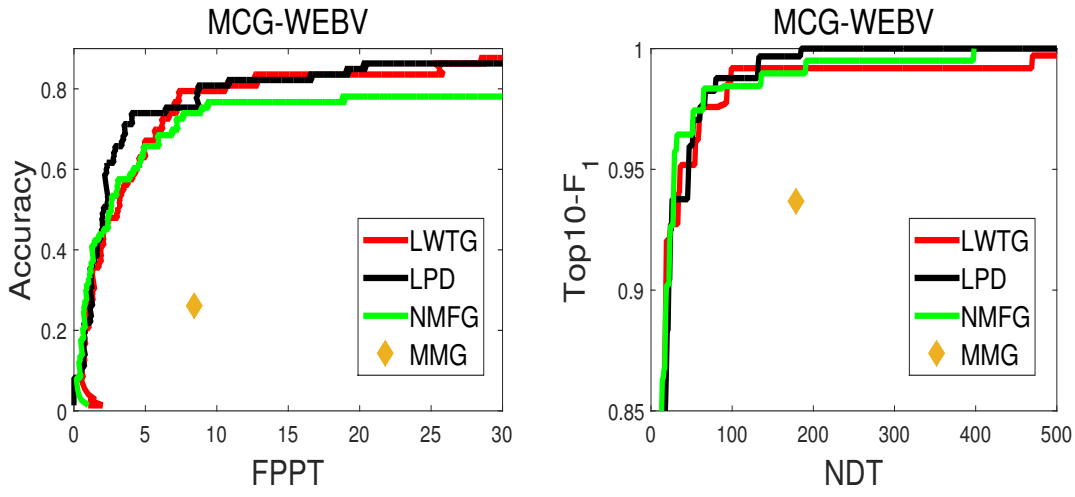
c) **Latent Poisson Deconvolution (LPD) [30]。** LPD算法在MCG-WEBV数据集和YKS数据集上均达到了当前最好的性能。相比单纯在PD上使用单一近邻图的方法，LPD利用多个近邻图来排序话题。这个算法证明了我们的算法可以在不利用多个近邻图的情况下达到可接受的结果。

为了尽可能公平的进行对比，在每个数据集上，我们使用了相同的实验设置。

1) **在MCG-WEBV数据集上进行网络话题生成：**图3.9展示在MCG-WEBV数据集上，我们的LWTG算法和当前几个优秀的网络话题检测算法的对比结果。从图中可以得出以下结论：

- 当FPPT大于8时，LWTG算法显著优于NMFG算法[49]，几乎能媲美LPD算法[30]。但是我们要注意到LPD算法使用非常耗时但是非常有效的NMG算法[28]，从两个k近邻图中生成话题。而NMFG算法[49]也使用NMG算法[28]来生成话题。作为对比，我们提出的LWTG算法避免去设计复杂的模型及其优化。
- 当FPPT小与8或者NDT大于200时，LWTG算法稍弱于NMFG算法和LPD算法。主要原因是泊松去卷积算法错误地将不正确的话题排在了正确话题的前

面。



(a) **(b)**

图 3.9 LWTG算法和其他网络话题检测算法在MCG-WEBV数据集上的对比

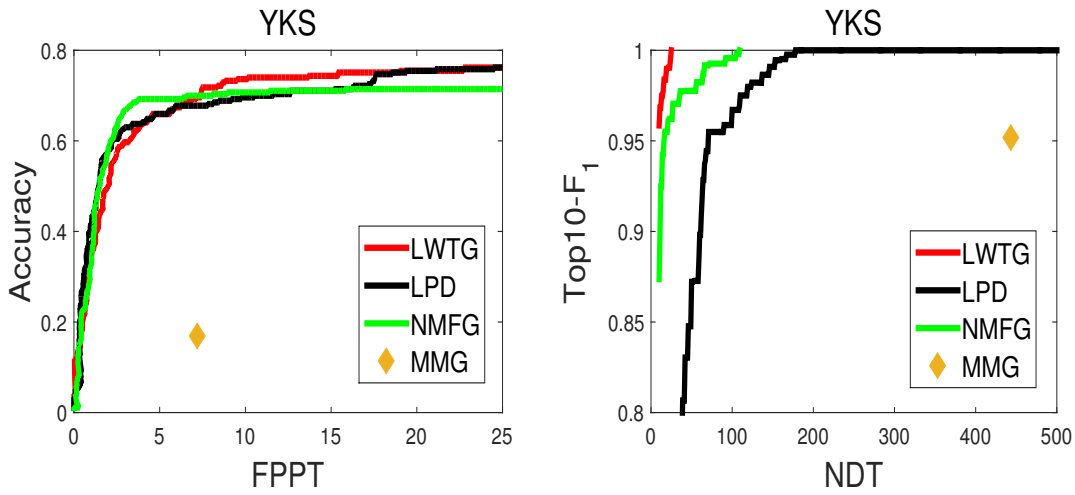
就话题生成效率来看，表3.3显示NMFG和LPD算法的运行时间分别是LWTG算法的71（如：5248/73）和158（如：11545/73）倍。这意味着我们的LWTG算法非常有希望能很好地对大规模网络数据进行快速的网络话题生成。与此同时，LWTG算法的话题检测的有效性也比得上当前最优秀的算法。

表 3.3 不同话题生成算法的运行时间（秒）和生成的话题数量对比

Data set(#topic)	LWTG(#topic)	LPD(#topic)	NMFG(#topic)	MMG(#topic)
MCG-WEBV(3660)	**73**(**2504**)	11545(7685)	5248(4238)	15(430)
YKS(8660)	**333**(7458)	29321(7524)	13847(**5714**)	252(445)

2) **在YKS数据集上进行网络话题生成:** YKS是一个跨平台的数据集，比MCG-WEBV数据集包含更多不同类型的话题，也更有挑战性。图3.10进一步说明了当FPPT大于8时，我们提出的LWTG算法仍然显著优于当前最好的话题检测算法。

同时就话题生成效率来看，表3.3显示了NMFG和LPD算法的运行时间分别是LWTG算法的41（如：13847/333）和88（如：29321/333）倍。这同样意味着我们的LWTG算法非常有希望能很好地对大规模网络数据进行快速的网络话题生成。虽然MMG算法的运行时间比较低，但是该算法对网页数量非常敏感。比



(a) **(b)**

图 3.10 LWTG算法和其他网络话题检测算法在YKS数据集上的对比

如说当网页数量从MCG-WEBV数据集的3,660增长到YKS数据集的8,660的时候，数据集只增长2.4（如：8660/3660）倍，但是，MMG算法的运行时间超过了16（如：252/15）倍。同时，MMG算法对于网络话题检测的效果也远远不如其他几个算法。

3.6 小结

本章介绍了一种面向网络数据的快速生成网络话题的算法，有望解决基于大规模网络数据的话题检测问题。该算法通过对网页计算其属于话题中心网页的概率来排序网页，进而生成不同规模的种子网页。然后模拟网络话题的Lévy Walks特性，定义网页-种子话题之间的相似度，采用一种基于种子话题的网页多分配算法来生成话题，同时使用层级阈值来截断话题，最终生成一系列过完备的话题。最后的实验从话题生成效果和话题生成效率两方面验证了我们提出的LWTG算法的优越性。

第4章　网络话题的快速排序

4.1　引言

信息技术和移动网络的快速发展，使得社交媒体极大地促进了用户生成式数据（User-Generated Content，UGC）[1]的产生和传播。海量的UGC数据使得用户难以从中快速获取热点话题。

由这个实际问题出发，许多网络话题检测算法[1, 30, 54]试图自动地将网页组织成有意义的话题。当前最好的网络话题检测算法是在相似度图上对话题的兴趣度进行排序[1, 49]。具体地说，泊松去卷积算法（Poisson Deconvolution，PD）通过扩散网页间的相似度来分配每个话题相应的权重[1]。虽然相似度图可以不仅可以通过在线k近邻图（k-Nearest Neighborhood Graph，k-N^{2}G）[38]构建，也能通过稀疏矩阵的方式存储。但是，一个严重的问题是PD算法在面对大规模网络数据时，无法进行高效地处理。这是因为PD算法迭代的每一轮必须使用所有的数据在内存中重构一个$N \times N$的浮点型矩阵，其中N是网页的数量。

那么，我们能否让PD算法在每次迭代的时候只使用一小部分样本呢？一个简单且有效的办法是随机优化[39]。这个办法能够带来至少两个好处：减少物理内存需求的同时，避免重构一个规模为$N \times N$的相似度图。然而，PD算法是通过期望最大化算法（Expectation-Maximization algorithm, EM）来优化的，它必须保持一个同相似度图一样规模的隐变量。

本章，我们提出了一个随机泊松去卷积算法（Stochastic Poisson Deconvolution，SPD）算法来处理大规模网络数据的话题检测问题。当每一轮迭代只能利用到一小部分样本时，SPD迭代地构建一个期望目标函数的代理函数。与此同时，只有一小部分随机采样的数据被用来更新代理函数。因此，通过避免把所有数据加载进内存，SPD算法显著地减少了运行时间。

据我们所知，这是第一个致力于解决网络话题检测中的泊松去卷积算法的可扩展性问题的算法。SPD算法不仅概念上简单而且实际上也非常有效。在一个大规模网络数据集中，SPD算法能够极大地缩短训练时间。比如在一个包含200,000个网页的数据集中，大概会有12.6倍的加速比。同时，在两个公开数据集上，SPD算法在话题检测效果上能够达到PD算法相同的水准。

4.2 泊松去卷积算法（PD）

先来回顾下泊松去卷积算法（Poisson Deconvolution，PD）[1]。

给定一系列网页集合$\mathcal{X} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$，我们将这些网页转换成相似度图$G = (V, E, A)$，其中$V$表示网页集合$\mathcal{X}$，仿射矩阵$A(a_{ij} \in A)$表示将网页之间经过尺度化和截断后的相似度值，边集合$E(e_{ij} \in E)$表示网页间相似度值为非0的边。具体如何构建相似度图可以参考论文[30]。

通过相似度图G，可以生成一系列多粒度的话题集合$C_k(k = 1, \ldots, K)$。一个话题C_k表示成：

$$C_k = c_k^\top \circ c_k \tag{4.1}$$

其中指示向量$c_k \in \{0, 1\}^{1\times N}$，$c_k$向量上第$i$位置上为1 或者 0 意味这话题$C_k$是否包含网页$w_i$。操作符$\circ$ 表示将矩阵$c_k^\top c_k$的对角元素设为0。

给定一系列话题集合$\{C_1, \ldots, C_K\}$和相似度图$G = (V, E, A)$，话题C_k的权重μ_k可以通过以下泊松噪声来学习：

$$\begin{aligned} r_{ij} &\sim \text{Poisson}(a_{ij}) \\ where\ \ r_{ij} &= \sum_{k=1}^{K} \mu_k C_{k_{ij}} \end{aligned} \tag{4.2}$$

话题的兴趣度定义为$i_k = \mu_k \cdot |C_k|$，其中$|C_k|$表示话题C_k所包含的网页数量。

通过应用EM算法，PD算法可以通过如下迭代优化得到解：

$$\mu_k = \frac{\sum_{a_{ij}\in C_k} a_{ij} P_{k_{ij}}}{\sum_{a_{ij}\in C_k} C_{k_{ij}}} \tag{4.3}$$

其中，$P_{k_{ij}}(\sum_{k=1}^{K} P_{k_{ij}} = 1)$是隐变量，并且，$P_{k_{ij}} = \frac{\mu_k C_{k_{ij}}}{\sum_{m=1}^{K} \mu_m C_{m_{ij}}}$[1]。

4.3 随机泊松去卷积算法（SPD）

由于PD算法无法高效地处理大规模网络数据，我们提出了SPD算法以实现PD算法在大规模网络数据上的可扩展性。算法4展示了SPD算法的处理过程。通过假设相似度图中的边是独立同分布于一个未知的分布，我们在第t轮迭代时，从相似度图中随机采样一小批边集合$\bar{A}^t$作为该轮训练样本。然而，实际上，这一小批边是在随机排列训练集样本后，通过循环抽取得到的。这是因为我们很难获得真正独立同分布的样本。

算法 4 随机泊松去卷积算法（SPD）

Input: 相似度图 $G = (V, E, A)$，话题$C_k(k = 1, ..., K)$，批样本数b，迭代次数T

Output: 话题权重$\boldsymbol{\mu}$

初始化累积中间值$\bar{B}^0 = \bar{D}^0 = \mathbf{0}^{K\text{x}1}$

初始化参数$W = \mathbf{0}^{K\text{x}1}$， β，α， $\boldsymbol{\mu}^0$

for $t = 1, ..., T$ **do**

从相似度矩阵G中随机选择几行，构成每轮训练样本：$\bar{A}^t$

通过公式(4.9)更新权重w_k

通过公式(4.6)计算临时变量B_k^t， D_k^t

更新累积中间值：

$$\bar{B}_k^t = (1 - w_k^t)\bar{B}_k^{t-1} + w_k^t B_k^t$$

$$\bar{D}_k^t = (1 - w_k^t)\bar{D}_k^{t-1} + w_k^t D_k^t$$

更新当前权重估计：$\mu_k^t = \frac{\bar{B}_k^t}{\bar{D}_k^t}$

end for

具体来说，PD算法公式(4.2)的目标函数等价于如下问题：

$$\begin{aligned} &\max \ln \prod_{a_{ij} \in \bar{A}^t} \text{Poisson}(a_{ij}) \\ \Leftrightarrow &\min \underbrace{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} \left(\sum_{k=1}^{K} \mu_k C_{k_{ij}} - a_{ij} \ln \sum_{k=1}^{K} \mu_k C_{k_{ij}} \right)}_{f^t(\bar{A}^t, \boldsymbol{\mu})} \end{aligned} \tag{4.4}$$

其中b是在相似度矩阵中这一小批样本$\bar{A}^t(\bar{A}^t \in \mathbb{R}^{b\text{x}N})$的行数。

使用Jensen不等式，公式(4.4)中的似然函数的上界被用来作为代理函数：

$$f^t(\bar{A}^t, \boldsymbol{\mu}) \leqslant \underbrace{\frac{1}{b} \sum_{a_{ij} \in \bar{A}^t} \left(\sum_{k=1}^{K} \mu_k C_{k_{ij}}^t - a_{ij} \sum_{k=1}^{K} P_{k_{ij}}^t \ln \frac{\mu_k C_{k_{ij}}^t}{P_{k_{ij}}^t} \right)}_{J^t(f^t, \boldsymbol{\mu}^{t-1})} \tag{4.5}$$

其中$C_{k_{ij}}^t$意味着与采样边相关的第k个话题，$P_{k_{ij}}^t (\sum_{k=1}^{K} P_{k_{ij}}^t = 1)$是第$t$轮迭代时的隐变量，并且$P_{k_{ij}}^t = \frac{\mu_k^{t-1} C_{k_{ij}}^t}{\sum_{k=1}^{K} \mu_k^{t-1} C_{k_{ij}}^t}$。这里，我们要注意到，并非每条边均能对所有话题产生影响。一般情况下，每条边只能影响到包含该边的少数几个话题。也

就是说，每一小批边样本只能影响一小部分话题，即只需更新该部分被影响的话题权重。

代理函数$J^t(f^t, \mu^{t-1})$关于μ_k的梯度为:

$$\frac{d}{d\mu_k}J^t(f^t, \boldsymbol{\mu}^{t-1}) = \underbrace{\frac{1}{b}\sum_{a_{ij}\in\bar{A}^t} C^t_{k_{ij}}}_{D^t_k} - \underbrace{\frac{\frac{1}{b}\sum_{a_{ij}\in\bar{A}^t} a_{ij}P^t_{k_{ij}}}{\mu_k}}_{B^t_k} \tag{4.6}$$

其中$D^t_k \in \mathbb{R}^{1x1}$和$B^t_k \in \mathbb{R}^{1x1}$是临时变量。

命题 4.1. （迭代更新过程）给定临时变量B^t_k和D^t_k，按如下方式迭代更新μ_k：

$$\mu^t_k = \frac{\bar{B}^t_k}{\bar{D}^t_k} \quad s.t.: \quad k \in \{k|\exists C^t_{k_{ij}} \neq 0\} \tag{4.7}$$

其中

$$\begin{aligned} \bar{B}^t_k &\leftarrow (1-w^t_k)\bar{B}^{t-1}_k + w^t_k B^t_k \\ \bar{D}^t_k &\leftarrow (1-w^t_k)\bar{D}^{t-1}_k + w^t_k D^t_k \end{aligned} \tag{4.8}$$

其中，第t轮迭代的权重w^t_k为：

$$w^t_k = \beta\sqrt{\frac{1+\alpha}{W_k+\alpha}} \tag{4.9}$$

其中$\beta \in (0,1]$，$\alpha \geq 0$，$W_k = W_k + 1$，k为那些受到影响的话题索引。

证明. 遵循论文SMM[46]的建议，近似代理函数和当前估计的结合得到如下公式：

$$\bar{J}^t_k \leftarrow (1-w^t_k)\bar{J}^{t-1}_k + w^t_k J^t_k \tag{4.10}$$

通过最小化公式(4.10)，可以得到$\mu^t_k = \arg\min \bar{J}^t_k(\mu)$。这个过程可以根据如下推导得到：

当$t = m$, 通过最小化公式(4.10)，同时利用公式(4.6)，(4.7)，(4.8)，我们可以得到：

$$\mu^m_k = \frac{(1-w^m_k)\bar{B}^{m-1}_k + w^m_k B^m_k}{(1-w^m_k)\bar{D}^{m-1}_k + w^m_k D^m_k} = \frac{\bar{B}^m_k}{\bar{D}^m_k} \tag{4.11}$$

最终，不失一般性，我们可以得到μ^t_k在第t轮的更新过程(4.7)，(4.8)，(4.9)。 □

权重参数w在近似代理函数公式(4.10)中扮演了指数加权移动平均的作用（Exponentially Weighted Moving Average，EWMA）。EWMA是一无限脉冲响应过滤器，代表了指数衰减的权重。注意到旧代理函数的权重呈现指数衰减但不会减少到0。权重w反映了当前代理函数的重要性，权重$1-w$反映了旧代理函数的重要性。因此，代理函数越旧，其对应的权重值w越小。

4.4 异步并行的随机泊松去卷积算法

由于网络数据规模庞大，导致SPD算法可能还是不够高效。同时，多核系统已经普遍存在。所以，通过设计一个异步策略来并行化SPD算法，我们提出了一种基于多核系统的异步并行的SPD算法（AsySPD）。

假设我们有p个可以访问一个共享内存的线程（进程），待求解的话题权重μ、话题更新次数W以及累积中间值$\bar{B}$和$\bar{D}$位于共享内存中。同时对于每个线程，都可以访问共享内存中的变量，并且可以随机选择训练集中的样本。我们还进一步假设对共享内存中的向量型变量进行一致性读操作。

算法5展示了我们的AsySPD算法。在t-th轮更新时，对于每个线程，均随机采样一批样本，并且进行独立计算和更新共享内存中的变量。M是所有线程在一轮数据集内更新的总次数。对于每个线程的每次独立更新，与算法4几乎一致，只是有些全局性变量（如话题更新次数W和累积中间值$\bar{B}$、$\bar{D}$）位于共享内存中，所有线程均可以对其进行更新。最后在更新共享内存中的$\boldsymbol{\mu}$时，使用$\boldsymbol{\mu_k} = \frac{\boldsymbol{\mu_k}+\boldsymbol{\mu}_k^m}{2}$公式来融合当前线程带来的更新。

算法 5 异步并行随机泊松去卷积算法（AsySPD）

初始化: p个线程，共享内存中的$\boldsymbol{\mu}$、W、$\bar{B}$、$\bar{D}$

Output: 共享内存中的$\boldsymbol{\mu}$

for $t = 1, ..., T$ **do**

对于每个线程，执行

for $m = 1, ..., M$ **do**

从相似度矩阵G中随机选择b行，构成每轮训练样本：$\bar{A}^m$

更新$W_k = W_k + 1$，其中k为受到样本$\bar{A}^m$影响的话题索引

通过公式(4.9)计算临时权重w_k^m

通过公式(4.6)计算临时变量B_k^m，D_k^m

通过公式(4.8)更新$\bar{B}_k$和$\bar{D}_k$

计算临时$\boldsymbol{\mu}_k^m = \frac{\bar{B}_k}{\bar{D}_k}$

更新$\boldsymbol{\mu_k} = \frac{\boldsymbol{\mu_k} + \boldsymbol{\mu}_k^m}{2}$

end for

end for

表 4.1 数据集的统计信息汇总

Data sets	#Webpage	#Topics	knn	Sparsity(%)
MCG-WEBV	3660	4240	100	2.73
YKS	8660	5252	20	0.273

4.5 实验验证

4.5.1 数据集、特征、评估标准、实验设置

数据集： 在本章实验中，我们仍然使用MCG和YKS两个数据集来评估我们提出的SPD算法。同时为了验证SPD算法在大规模网络数据上的效率，我们通过在对角位置叠加50个MCG-WEBV数据集的相似度矩阵，最终得到一个大约包含200,000个网页的人工数据集的相似度矩阵。两个真实数据集的统计信息如表4.1所示。

特征： 在数据预处理阶段，仍然对MCG和YKS数据集提取文本数据的TF-IDF特征。然后使用余弦距离计算网夜间的相似度。接着保留与每个网页最相似

的k个网页的相似度值，其余置为0，最终生成一个k近邻的相似度图。

评估标准： 仍然使用Top10-F_1 v.s. Number of Detected Topics (NDT)[23]和 $Accuracy$ v.s. False Positive Per Topic ($FPPT$)[1]。当两个方法有相同的Top10-F_1或$Accuracy$分数后，拥有更小NDT或者$FPPT$的算法有更好的性能。同时，对于算法运行时间效率方面的评估，我们采用了目标函数的收敛曲线。

实验设置： 所有实验使用了相同的实验设置，比如相似度图和待排序话题集合。所有实验代码使用python语言在3.6G Hz CPU和32G内存的电脑上运行。

4.5.2 复杂度分析

对PD算法，需要在内存中分配一个$N \times N$的浮点型矩阵来存储重构的相似度图。因此，PD算法的空间复杂度和时间复杂度分别为$O(N^2)$ and $O(TN^2)$。作为对比，SPD算法只需要在内存中保存一个规模为$b \times N$的浮点型矩阵，其中$b \times N$表示每次采样的一小批边样本规模（比如，$\bar{A}^t \in \mathbb{R}^{b \text{xN}}$）。所以SPD算法的空间复杂度为$O(bN)$ $(b \ll N)$。

4.5.3 在人工数据集上对比PD和SPD

为了评估SPD算法的运行效率，我们构造了一个较大规模的人工数据集。本章中，如果一个算法前后两轮迭代的目标似然函数的对数值的变化小于10^{-5}，那么我们认为该算法收敛到了一个局部极小值。同时定义算法A对算法B的加速比是算法收敛到局部极小值时的CPU运行时间之间的比值，如公式(4.12)：

$$speedup = \frac{\text{算法A收敛到局部极小值时的CPU运行时间}}{\text{算法B收敛到局部极小值时的CPU运行时间}} \tag{4.12}$$

图4.1表明，就收敛速度而言，SPD算法显著由于PD算法。在我们的评估中，SPD算法相比PD算法会有大概12.6倍的加速比。而且，在每一轮迭代，虽然PD算法的目标函数下降得更多，但是我们的随机方法相比批处理方法使用更少的训练时间。这是因为在每一轮迭代更新的时候，PD算法使用所有的数据去进行一个精确的更新，而SPD算法只使用一小部分样本去进行一个近似的更新。当训练一个大规模的网络数据时，由于计算效率和内存限制的原因，PD算法将会比SPD算法花费更多的时间来使用一轮数据集进行更新。因此，在处理大规模网络数据时，SPD算法的收敛速度会比PD算法快很多。这同时也验证了公式(4.10)作为损失函数的优点：一个算法无需花费太多的时间去精确地最

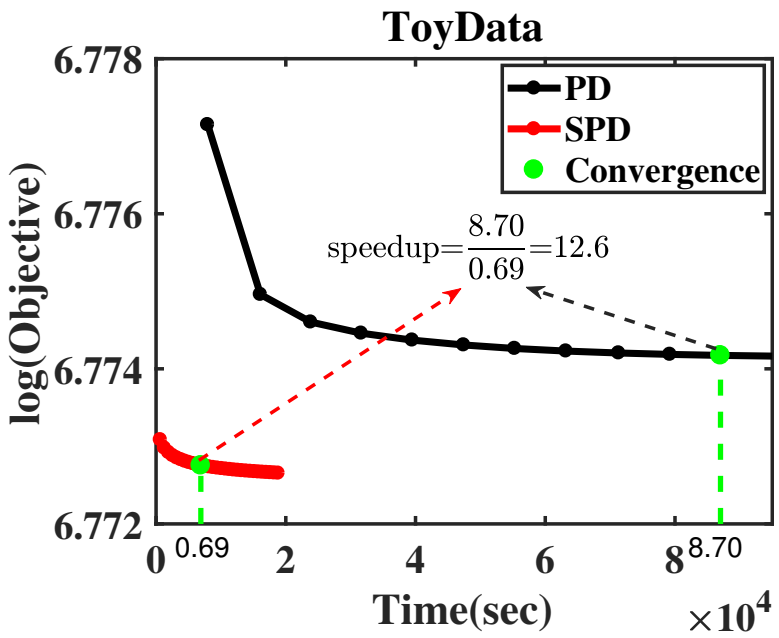


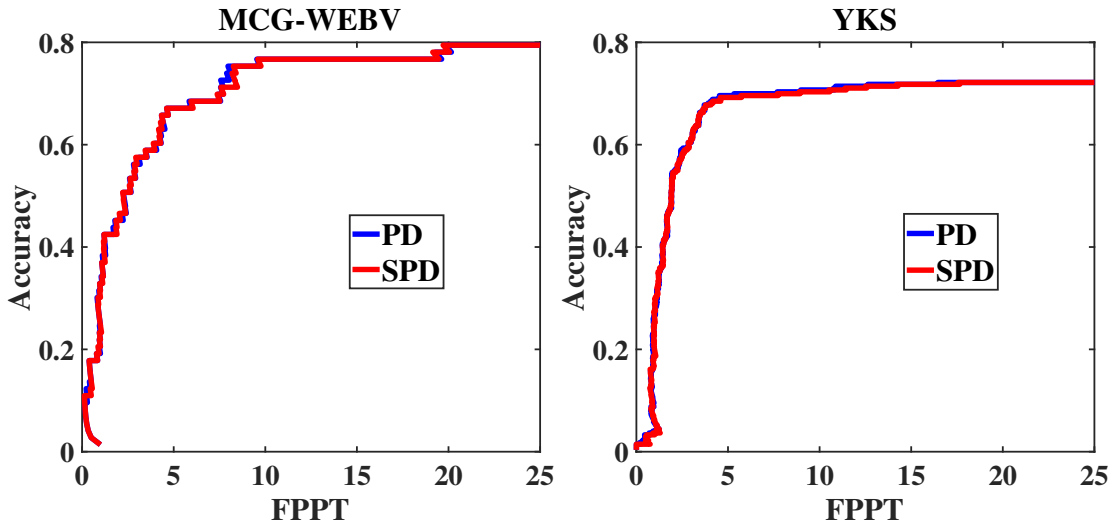
图 4.1 PD和SPD算法在ToyData上的收敛对比

小化经验损失。另外，就像梯度下降一样，PD算法同样遭遇局部极小值问题，而SPD算法可以更好地避免该问题，从而能够收敛到一个更小的局部极小值。

4.5.4 在MCG-WEBV和YKS上对比PD和SPD

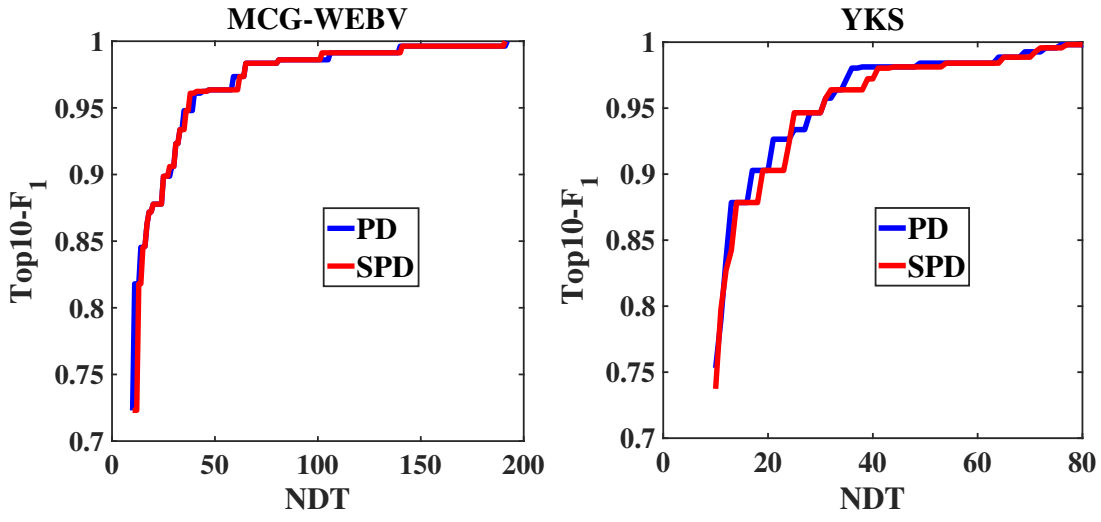
对于话题排序效果，图4.2使用*Accuracy* v.s. *FPPT*在MCG-WEBV和YKS数据集上对比了PD算法和SPD算法的结果。从图4.2可以看出SPD算法的话题排序效果与PD算法相似。同时图4.3进一步使用Top10-F_1 v.s. *NDT*证明了SPD算法的排序效果不比PD算法差。综上，说明我们的SPD算法可以达到和PD算法相似的话题排序效果。

对于话题排序效率，图4.4对比PD算法和SPD算法在MCG-WEBV和YKS数据集上的收敛情况。从这两幅图中可以看出SPD算法可以收敛到一个更低的对数似然函数值。这意味这SPD算法可以在保证算法排序效果的同时使目标函数收敛到更低的值。但是由于这两个数据集都很小，SPD算法的随机采样优势没能发挥出来，所以导致SPD算法训练一遍数据集花费的时间比PD算法多。所以在处理小规模网络数据时，SPD算法收敛会比PD算法慢。但是，在处理大规模网络数据时，SPD算法的随机优势就能得以发挥使其收敛速度比PD算法快。



(a) (b)

图 4.2 使用*Accuracy* v.s. *FPPT*对比PD和SPD算法



(a) (b)

图 4.3 使用Top10-F_1 v.s. *NDT*对比PD和SPD算法

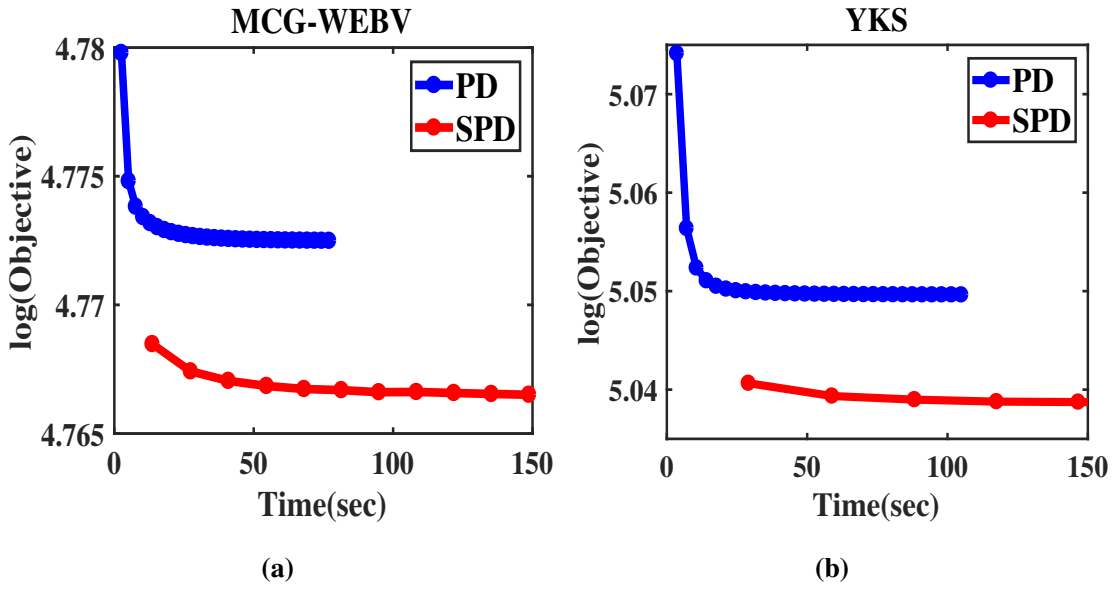


图 4.4 使用目标函数收敛曲线对比PD和SPD算法

4.5.5 参数分析

4.5.5.1 批样本数量b：

图4.5展示了在MCG-WEBV数据集上，不同的批样本数量b的大小对SPD算法目标函数收敛曲线的影响。从中可以看出不同的批样本数量b均收敛到一个局部极小值。此外，b值越小，对数目标函数值也会收敛到更低的程度。一个可能的解释是更小的批样本数量会带来更大的随机性，使得SPD算法更有可能逃离较差的局部极小值，到达一个相对较好的局部极小值。

有趣的是，图4.6显示不同大小的批样本数量均能获得相似的*Accuracy*和Top10-F_1值。可能的原因是即使不同的b值导致最终的话题权重μ值的分布不同，对数目标函数值不同。但是，只要这些话题权重mu值的相对大小分布几乎不变，即对μ值排序后的所处位置在不同批样本数量时几乎不变，那么就有可能达到相似的*Accuracy*和Top10-F_1。因此，批样本数量b的大小不会影响SPD算法的话题排序性能。

4.5.5.2 权重参数α和β：

我们在公式(4.9)使用了一个衰减权重。其中β是一个初始权重，α是一个权重衰减因子。

图4.7展示了当固定其他参数（比如：$\alpha = 0$，$b = 128$，$T = 30$），不同β值在收敛速度上的对比。正如所预期的那样，β越大，SPD算法的收敛速度越快。这

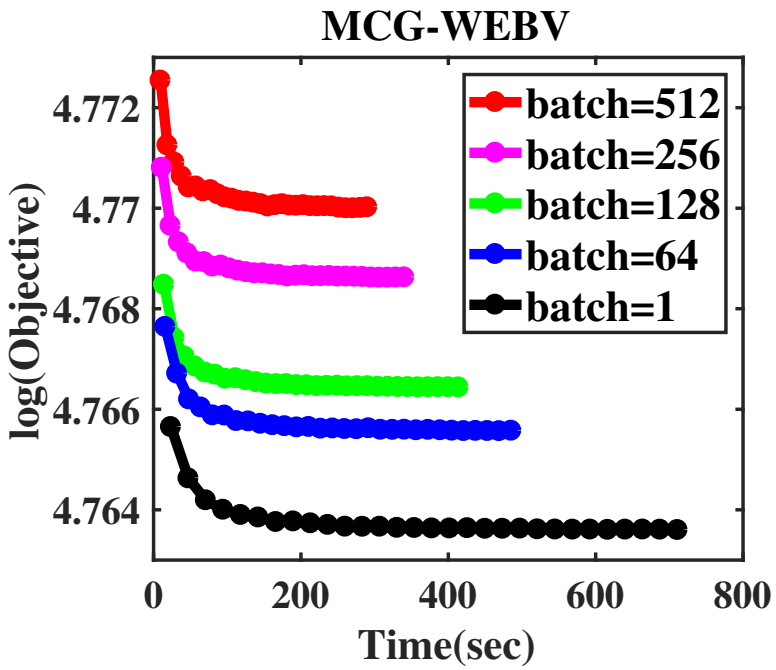
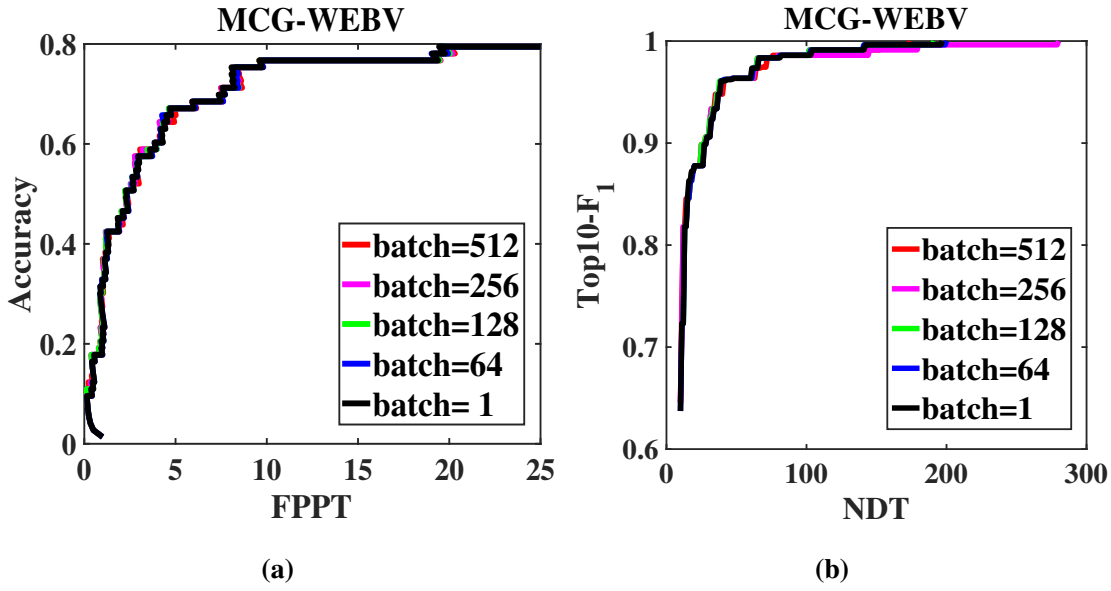


图 **4.5** 不同批样本数量*b*对目标函数收敛曲线的影响



(a) **(b)**

图 **4.6** 不同批样本数量*b*的话题排序效果对比

是因为一个较大的β值使得近似代理函数能够快速适应当前最新的代理函数。

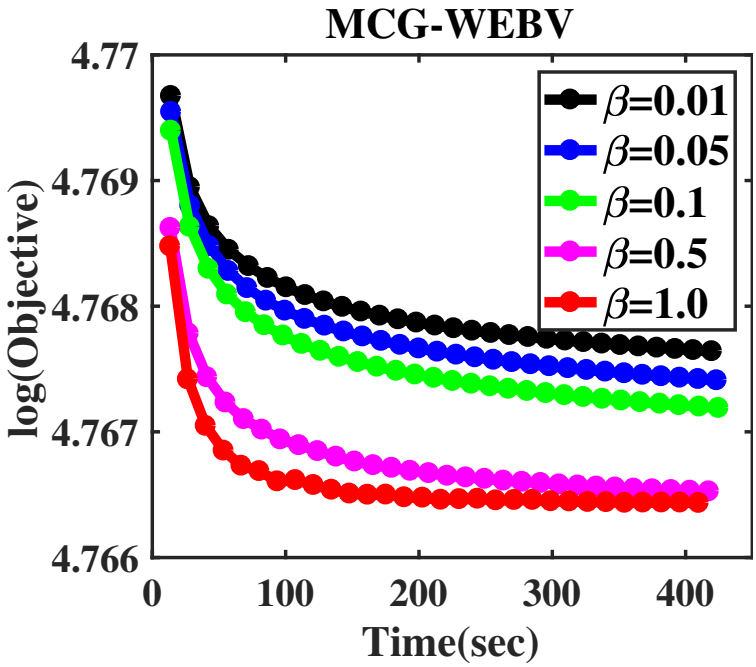
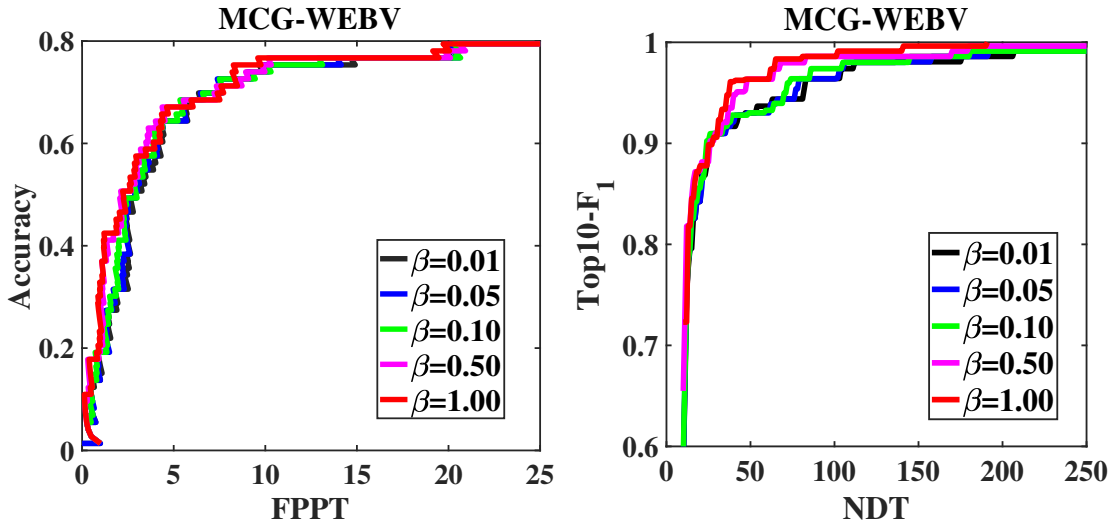


图 4.7 不同β值对目标函数收敛曲线的影响

从图4.8的结果中，我们发现一个较大的β值不仅会带来更快的收敛速度，而且也会带来更好的话题排序效果。



(a) (b)

图 4.8 不同β值的话题排序效果对比

图4.9展示了当固定其他参数（比如：$\beta = 1$，$b = 128$，$T = 30$），不同α值在收敛速度上的对比。从图中可以看出一个更小的α会带来一个更平滑的目标函数

收敛曲线。这是因为更小的α值不仅使得β值衰减更快，而且也会使得SPD算法对于当前样本带来的代理函数更加的稳定。虽然不同α值的收敛过程不太一样，但是最终都会收敛到一个相似的局部极小值。

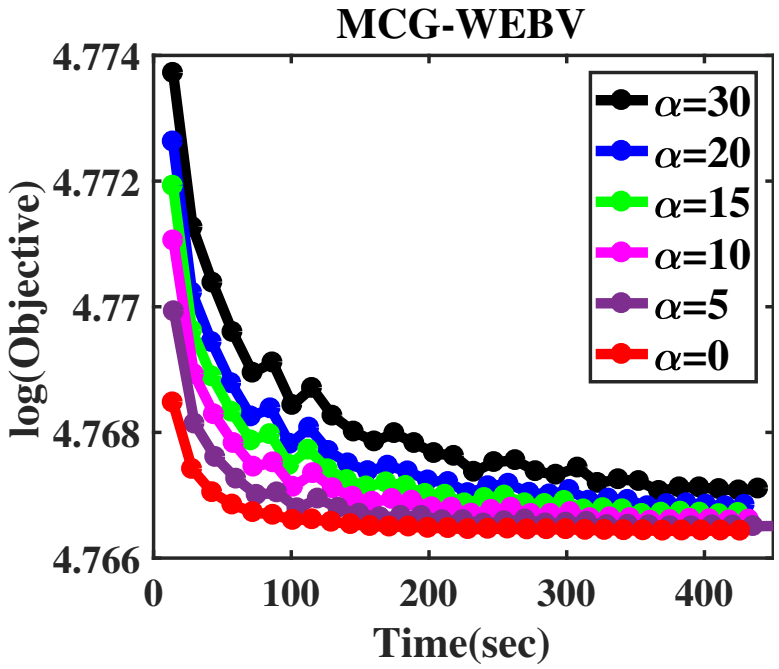


图 4.9 不同α值对目标函数收敛曲线的影响

从图4.10的结果中，可以看出不同α值的话题排序性能几乎是一致的。

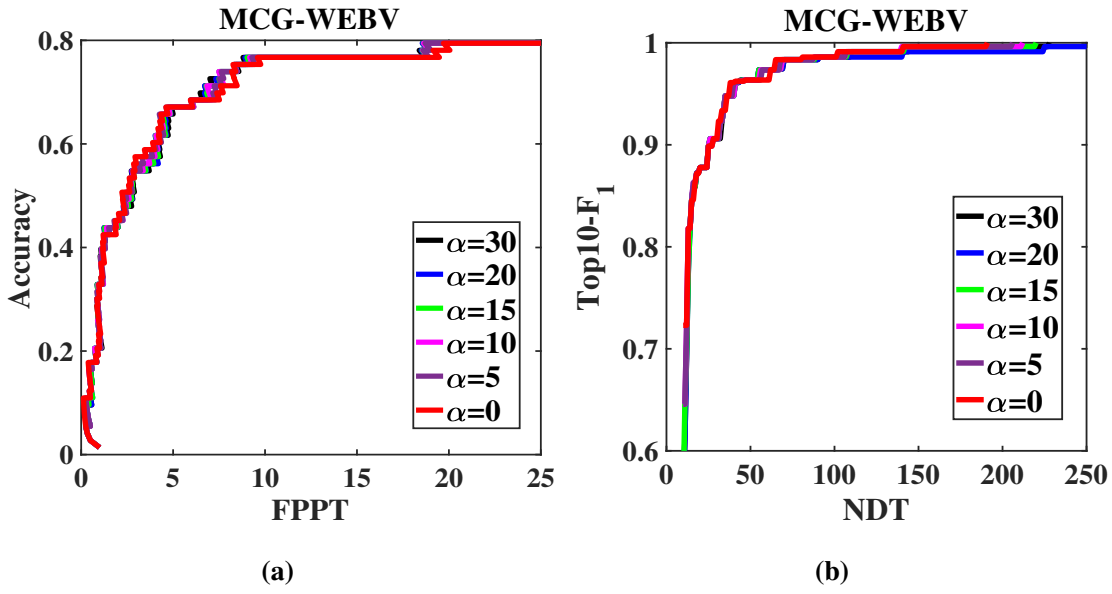


图 4.10 不同α值的话题排序效果对比

通过以上实验结果可以得到，不同的α和β值虽然会在一定程度上影响SPD算

法中目标函数的收敛速度，但是从*Accuracy*和Top10-F_1来看，这些不同参数的不同取值对话题排序效果的影响不是很大。所以，可以认为我们的SPD算法是一个对参数具有鲁棒性的算法。

4.5.6 异步并行实验

我们使用python语言开了4个进程来实现AsySPD算法。图4.11展示了SPD算法和AsySPD算法在人工数据集上的收敛情况。从图中可以看出，AsySPD算法相对SPD算法的加速比达到了3.3倍，证明AsySPD算法是有效的。但是AsySPD算法的收敛值比SPD算法大，这可能是因为使用了异步无锁并行策略，导致在使用相同数量样本进行更新时，样本利用率不高，且存在更新冲突，从而收敛到一个较大的局部极小值。

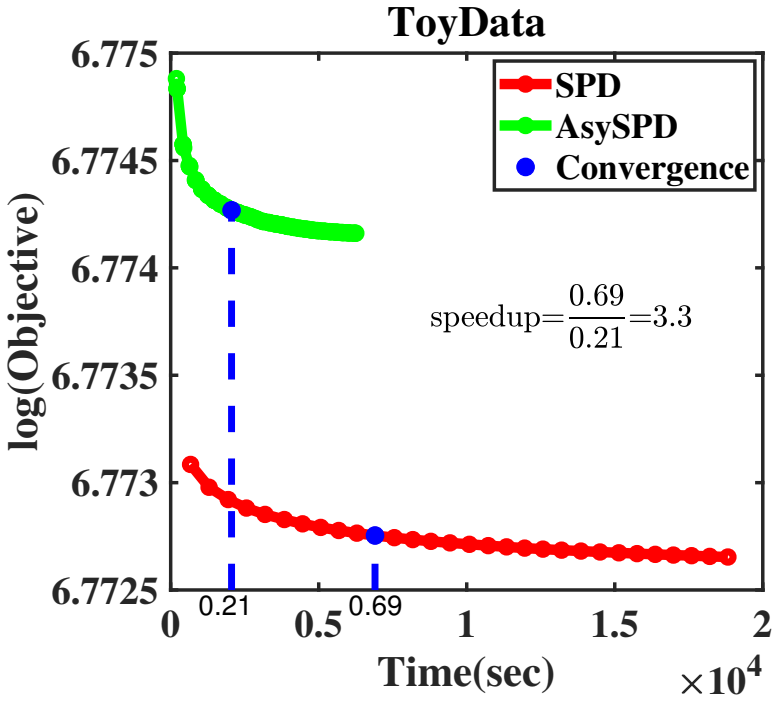


图 4.11 SPD和AsySPD算法在ToyData上的收敛对比

图4.12展示了SPD算法和AsySPD算法在MCG-WEBV和YKS数据集上的收敛情况。同样发现AsySPD相比SPD收敛到一个较大的局部极小值。AsySPD算法的收敛趋势比较接近PD算法。

最后，我们在MCG-WEBV和YKS数据集上对比这两个算法的话题排序效果。从图4.13可以看出AsySPD算法与SPD算法在*Accuracy*上的排序上效果一致。从图4.14可以看出这两个算法在Top10-F_1的排序上效果也几乎一致。综上，实际

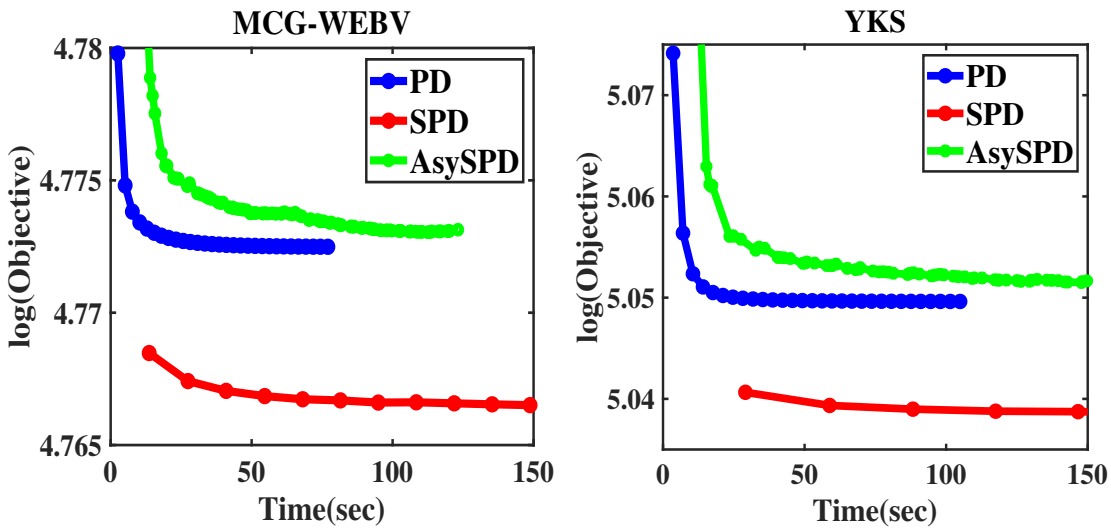


图 4.12 SPD和AsySPD算法在MCG-WEBV和YKS数据集上的收敛对比

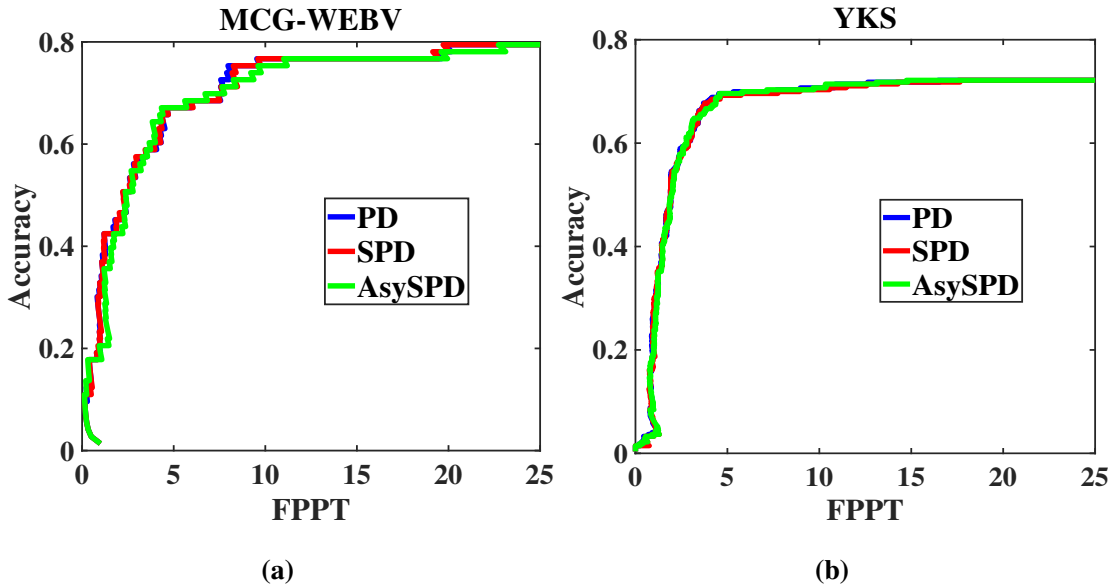


图 4.13 使用*Accuracy* v.s. *FPPT*对比SPD和AsySPD算法

数据集的实验结果可以经验地证明我们AsySPD算法的有效性和高效性。

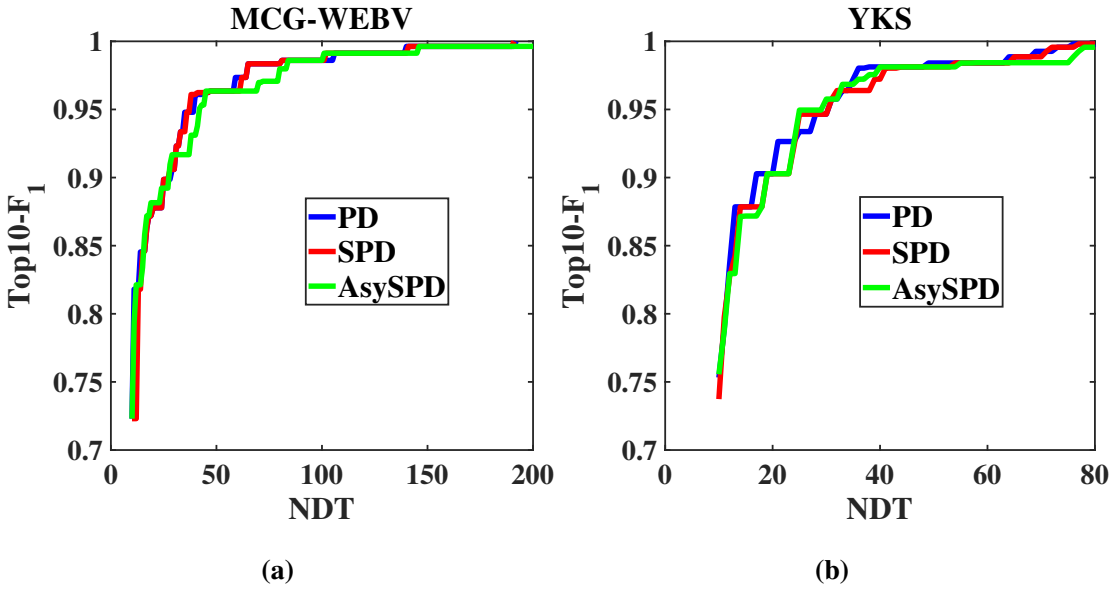


图 4.14 使用Top10-F_1 v.s. NDT对比SPD和AsySPD算法

4.6 小结

在本章中，我们提出了一种SPD算法，将随机优化最小化原则应用到PD算法中，从而能够优雅地处理大规模网络数据的话题检测问题。SPD算法迭代的利用小批量样本来更新目标函数的上界代理函数，同时最小化该代理函数，从而单调地驱使目标函数值下降。应用过程中注意到话题检测的特殊场景：一小批样本只能更新一部分话题的权重，而传统的随机梯度下降中的小批量样本能更新所有的权重。所以，在每次更新时，只更新那些被当前批样本影响到的话题。最终在话题检测效果方面，我们通过实验展示了我们的算法可以取得和当前最好的话题检测算法相似的性能。同时，我们构造了一个较大规模的人工数据集来验证我们算法高效的收敛速度。最后，我们提出AsySPD算法来实现SPD算法的异步并行更新。

第5章　总结与展望

5.1　本文工作总结

本文致力于大规模网络数据的话题检测研究。基于多粒度网络话题的无监督排序算法，将话题检测问题转换为无监督的排序问题。本文提出两个算法来尽可能地解决大规模网络数据带来的效率问题。这两个算法分别针对网络话题的快速生成和网络话题的快速排序，并且在MCG-WEBV和YKS数据集上验证了我们算法效果和效率。同时我们还构造了一个较大规模的网络数据集来进一步验证我们的算法在大规模网络数据上的效率。

在网络话题的快速生成中，我们首先分析了网络话题在相似度空间上的统计模式，发现在相似度空间上的网络话题与Lévyy Walks存在统计意义上的相似性，就是说一个话题内的所有网页之间的相似度服从一个未知参数的重尾分布。同时发现网络话题并不完全满足普遍所接受的看似合理的假设：话题内任意一个网页与话题内其他网页之间的类内相似度应该大于该网页与话题外其他网页之间的类间相似度。据此，我们提出了一种基于Lévy Walks的话题生成算法：首先使用余弦距离构造网页之间的相似度图，并通过使用k近邻算法保留与网页最相似的k个网页，再使用阈值截断该相似度图，从而过滤大部分噪声网页；然后，我们通过站点熵率来评估每个网页成为话题中心网页的概率，并使用贪心算法来找到小规模的种子网页，初始化种子网页为种子话题；接着我们将网页带来的平均相似度与话题平均相似度的比值定义为网页与话题的相似度，然后基于该相似度将非噪声网页分配给多个种子话题。最后在分配时考虑当前话题的平均相似度，使用层级阈值进行截断，从而产生多粒度的网络话题。该算法简单高效，在MCG-WEBV和YKS数据集上，针对话题的召回率超过了当前最好的基于随机游走的非负矩阵分解算法，同时还极大的提高了效率。

在网络话题的快速排序中，我们首先回顾了PD算法，并分析其在处理大规模数据时的不足。然后由于基于EM算法来优化的PD算法不能够直接使用随机梯度下降来优化，所以我们找到了优化最小化原则，这是EM的泛化版本。接着我们使用随机优化最小化原则来优化PD算法，提出了SPD算法。该算法每轮迭代仅仅利用一小批相似度边来更新目标函数的上界代理函数，然后优化求解当

前估计。由于每轮迭代仅仅利用一小批样本，使得SPD减少了物理内存的需求，同时提高了计算效率。在MCG-WEBV和YKS数据集上的实验验证SPD算法在话题检测方面的效果，同时在构造的人工数据集上的实验验证了SPD算法的效率。此外，基于多核系统，我们还实现了该算法的异步并行版本-AsySPD算法。

5.2 未来研究展望

随着社交媒体与移动网络的进一步发展与普及，人们会越来越多地投入到网络数据的创作和传播。因此，网络数据的规模只会越来越庞大，其中蕴含的有价值的信息也会越来越多。虽然本文提出的两个行之有效的算法比当前最好的网络话题检测算法效率更高，更具有可扩展性，但是离真正能处理超大规模网络数据，还有很长一段路，我们提出的算法只是在这条路上迈进了一步。因此，面向大规模网络数据的话题检测研究仍将是一项任重道远的任务。在未来的研究中，我们将着重从以下几点展开：

(1) 在寻找种子网页的算法中，如果能找到一种更有效的评估网页作为话题中心网页的可能性，那么应该能进一步地提高话题的召回率；

(2) 如果能找到一种更有效更合适的方法来度量网页跟话题的相似度，那么可能会进一步提高话题生成效率和效果；

(3) 研究更高级的排序算法来提高LWTG算法所生成的话题的准确性；

(4) 基于现实场景下的数据是信息流模式的情况，将在线学习用于网络话题检测是另一项有趣的研究。

参考文献

[1] J.PANG, F.JIA, C.ZHANG, et al. Unsupervised web topic detection using a ranked clustering-like pattern across similarity cascades[J]. IEEE Transactions on Multimedia, 2015, 17(6): 843-853.

[2] J.ALLAN, J.CARBONELL, G.DODDINGTON, et al. Topic detection and tracking pilot study final report[C]//In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. 1998: 194-218.

[3] J.ALLAN. Topic detection and tracking: Event-based information organization[M]. 2002.

[4] W.X.ZHAO, J J, J.WENG, et al. Comparing twitter and traditional media using topic models [C]//Advances in Information Retrieval. 2011: 338-349.

[5] T.PIERCE Y, J.CARBONELL. A study of retrospective and on-line event detection[C]// Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1998: 28-36.

[6] Y.ZHAI, M.SHAH. Tracking news stories across different sources[C]//Proceedings of the 13th Annual ACM International Conference on Multimedia. 2005: 2-10.

[7] A.PONS-PORRATA, R.BERLANGA-LLAVORI, J.RUIZ-SHULCLOPER. Topic discovery based on text mining techniques[J]. Information Processing & Management, 2007, 43(3): 752-768.

[8] A.BOJCHEVSKI, Y.MATKOVIC, S.GüNNEMANN. Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 737-746.

[9] S.MAURUS, C.PLANT. Skinny-dip: Clustering in a sea of noise[C]//2016: 1055-1064.

[10] D.BLEI, M.DAVID, A.NG, et al. Latent dirichlet allocation[J]. Journal of machine learning research, 2003, 3:993-1022.

[11] J.CAO, Y.ZHANG, Y.SONG, et al. Mcg-webv: A benchmark dataset for web video analysis [J]. 2009, 10.

[12] Y.ZHANG, G.LI, L.CHU, et al. Cross-media topic detection: A multi-modality fusion framework[C]//2013 IEEE International Conference on Multimedia and Expo (ICME). 2013: 1-6.

[13] G.SALTON, C.BUCKLEY. Term-weighting approaches in automatic text retrieval[J]. Inf. Process. Manage., 1988, 24(5):513-523.

[14] S.DEERWESTER, S.T.DUMAIS, G.W.FURNAS, et al. Indexing by latent semantic analysis

[J]. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE, 1990, 41 (6):391-407.

[15] T.HOFMANN. Probabilistic latent semantic analysis[C]//Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. 1999: 289-296.

[16] Y.W.T, M.I.JORDAN, M.J.BEAL, et al. Hierarchical dirichlet processes[J]. Journal of the American Statistical Association, 2006, 101(476):1566-1581.

[17] D.BLEI, J.LAFFERTY. A correlated topic model of science[J]. Annals of Applied Sciences, 2007, 1:17-35.

[18] D.PUTTHIVIDHY, HT.ATTIAS, SS.MAGARAJAN. Topic regression multi-modal latent dirichlet allocation for image annotation[C]//Computer Vision and Pattern Recognition: volume 1. 2010: 3408-3415.

[19] S.PAPADOPOULOUS, C.ZIGKOLIS, Y.KOMPATSIARIS, et al. Cluster-based landmark and event detection on tagged photo collections[J]. IEEE Multimedia, 2011, 18(1):52-63.

[20] Y.JIA, M.SALZMANN, T.DARRELL. Learning cross-modality similarity for multinomial data[C]//2011 International Conference on Computer Vision. 2011: 2407-2414.

[21] S.OH, S.MCCLOSKEY, I.KIM, et al. Multimedia event detection with multimodal feature fusion and temporal concept localization[J]. Machine Vision and Applications, 2014, 25(1): 49-69.

[22] X.WU, G.HAUPTMANN, C.NGO. Novelty detection for crosslingual news story with visual duplicates and speech transcripts[C]//ACM Multimedia. 2007: 168-177.

[23] J.CAO, C.NGO, Y.ZHANG, et al. Tracking web video topics: Discovery, visualization, and monitoring[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21 (12):1835-1846.

[24] LM.AIELLO, G.PETKOS, C.MARTIN, et al. Sensing trending topics in twitter[J]. IEEE Transactions on Multimedia, 2013, 15(6):1268-1282.

[25] C.WANG, M.ZHANG S, L.RU. Automatic online news issue construction in web environment [C]//Proceedings of the 17th International Conference on World Wide Web. 2008: 457-466.

[26] H.LIU, S.YAN. Robust graph mode seeking by graph shift[C]//Proceedings of the 27th International Conference on International Conference on Machine Learning. 2010: 671-678.

[27] W.XU, X.LIU, Y.GONG. Document clustering based on non-negative matrix factorization [C]//Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. 2003: 267-273.

[28] Z.YANG, T.HAO, O.DIKMEN, et al. Clustering by nonnegative matrix factorization using graph random walk[M]//Advances in Neural Information Processing Systems 25. 2012: 1079-1087.

[29] Z.LI, J.LIU, S.CHEN, et al. Noise robust spectral clustering[C]//2007 IEEE 11th International Conference on Computer Vision. 2007: 1-8.

[30] J.PANG, F.TAO, C.ZHANG, et al. Robust latent poisson deconvolution from multiple features for web topic detection[J]. IEEE Transactions on Multimedia, 2016, 18(12):2482-2493.

[31] J.CHEN, K.LI, J.ZHU, et al. Warplda: a cache efficient o(1) algorithm for latent dirichlet allocation[J]. Proceedings of the Vldb Endowment, 2015, 9(10):744-755.

[32] Y.WANG, H.BAI, M.STANTON, et al. Plda: Parallel latent dirichlet allocation for large-scale applications[C]//Algorithmic Aspects in Information and Management: volume 5564. 2009: 301-314.

[33] Z.LIU, Y.ZHANG, EY.CHANG, et al. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing[J]. ACM Trans. Intell. Syst. Technol., 2011, 2(3):26:1-26:18.

[34] T.PERKINS, E.FOXALL, L.GLASS, et al. A scaling law for random walks on networks[J]. Nature communications, 2014, 5:5121.

[35] G.M.VISWANATHAN, F.BARTUMEUS, S.V.BULDYREV, et al. Levy flight random searches in biological phenomena[C]//volume 314. 2001: 208-213.

[36] I.RHEE, M.SHIN, S.HONG, et al. On the levy-walk nature of human mobility[J]. IEEE/ACM Transactions on Networking, 2011, 19(3):630-643.

[37] G.LI, S.D.S.REIS, A.A.MOREIRA, et al. Towards design principles for optimal transport networks[J]. Phys. Rev. Lett., 2010, 104:018701.

[38] T.DEBATTY, P.MICHIARDI, W.MEES. Fast online k-nn graph building[J]. CoRR, 2016.

[39] LA.HANNAH. Stochastic optimization[J]. International Encyclopedia of the Social and Behavioral Sciences, 2015, 5(5):473-481.

[40] NL.ROUX, M.SCHMIDT, F.BACH. A stochastic gradient method with an exponential convergence rate for finite training sets[C]//International Conference on Neural Information Processing Systems: volume 2. 2012: 2663-2671.

[41] R.JOHNSON, T.ZHANG. Accelerating stochastic gradient descent using predictive variance reduction[C]//International Conference on Neural Information Processing Systems: volume 1. 2013: 315-323.

[42] A.NITANDA. Stochastic proximal gradient descent with acceleration techniques[C]// International Conference on Neural Information Processing Systems: volume 2. 2014: 1574-1582.

[43] K.LANGE, DR.HUNTER, I.YANG. Optimization transfer using surrogate objective functions [J]. Journal of Computational and Graphical Statistics, 2000, 9(1):1-20.

[44] MJ.WAINWRIGHT, MI.JORDAN. Graphical models, exponential families, and variational inference[J]. Foundations and Trends® in Machine Learning, 2008, 1(1–2):1-305.

[45] A.BECK, M.TEBOULLE. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. Siam Journal on Imaging Sciences, 2009, 2(1):183-202.

[46] J.MAIRAL. Stochastic majorization-minimization algorithms for large-scale optimization [C]//International Conference on Neural Information Processing Systems: volume 2. 2013: 2283-2291.

[47] D.SHAHAF, C.GUESTRIN. Connecting the dots between news articles[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010: 623-632.

[48] Y.LIU, A.NICULESCU-MIZIL, W.GRYC. Topic-link lda: Joint models of topic and author community[C]//2009: 84.

[49] J.PANG, F.TAO, L.LI, et al. A two-step approach to describing web topics via probable keywords and prototype images from background-removed similarities[J]. Neurocomputing, 2018, 275:478-487.

[50] Z.S.HARRIS. Distributional structure[J]. WORD, 1954, 10(2-3):146-162.

[51] K.P.BURNHAM, D.R.ANDERSON. Multimodel inference: Understanding aic and bic in model selection[J]. Sociological Methods & Research, 2004, 33(2):261-304.

[52] M.LIU, O.TUZEL, S.RAMALINGAM, et al. Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(1):99-112.

[53] W.WANG, Y.WANG, Q.HUANG, et al. Measuring visual saliency by site entropy rate[C]// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010: 2368-2375.

[54] Y.ZHANG, G.LI, L.CHU, et al. Cross-media topic detection: a multi-modality fusion framework[C]//2013 IEEE International Conference on Multimedia and Expo (ICME). 2013: 1-6.

作者简介

作者简介：

姓名：林尽忠　性别：男　出生日期：1992.2.27　籍贯：福建省古田县

2012.9-2016.6 在杭州电子科技大学通信工程学院获得学士学位

2016.9-2019.6 在中国科学院大学计算机科学与技术学院攻读硕士学位

已发表(或正式接受)的学术论文：

[1] Jinzhong Lin, Junbiao Pang, Li Su, Yugui Liu, Qingming Huang, "Accelerating Topic Detection on Web for a Large-Scale Data Set via Stochastic Poisson Deconvolution", in Proceedings of International Conference on Multimedia Modeling, 2019, pp. 590-602.

获奖情况：

[1] 2019年被评为中国科学院“三好学生”

致　谢

时光荏苒，仿若白驹过隙。犹记得三年前独自一人来国科大面试，那情景，仿佛还发生在昨日，现在却到了要说再见的时候。这即将结束的三年北漂生活，同时也代表着学生生涯的结束。回望二十载的辛苦求学路，有近十载是独自异地求学，个中滋味，难以言表。在国科大的三年时光里，我不仅收获了很多知识，也得到了能力的提高和心理素质的锻炼。而这些，都离不开老师们的谆谆教诲，同学们的热情帮助以及家人朋友们的默默支持。在此，对你们致以衷心的感谢和祝福。

感谢我的父母。你们尽自己做大的努力给我提供良好的生活条件和求学环境，只愿我有更好的选择。从小到大，无论我做什么决定，你们总是无条件的支持我，鼓励我。相比学习成绩，更在乎我是否健康快乐。每每看到你们疲惫操劳的身影，我总是一阵心酸。只言片语无法表达我对你们的感谢和爱。祝愿二老身体健康，幸福快乐。

感谢黄庆明教授。在生活上对学生照顾有加，在科研上提供一流的设备，使我们能够心无旁骛地潜心科研。感谢您在面试阶段录取了我，给我打开一扇走入科研生活的大门，让我感受到学术的魅力。您严谨的科研态度、热情的关怀都使我铭记于心。在此也祝您身体健康，桃李满天下！感谢马丙鹏副教授，三年前，是您把懵懂的我招进中国科学院大学读研，从此走上科研之路，开始人生的新征程。感谢刘玉贵副教授在这三年对我的照顾，您严谨认真的治学态度和谦和豁达的人生态度着实令我钦佩。再次感谢马丙鹏老师和刘玉贵老师，祝您二位身体健康，事业顺利！

感谢庞俊彪副教授。感谢您这几年对我的指导和帮助。作为我的直接负责老师，您言传身教，真正做到了传道、授业、解惑。三年来，是您把我从一个科研门外汉，一步步带到门内。从课题的选择到算法研究，从实验开展到论文撰写，这其中的每一步都有您亲身参与，亲自指导。让我少走了很多弯路，同时也收获了很多。生活上，您对学生无微不至的体贴关怀；科研上，您对学生耐心有加，逐步指导。积极推动学生奋发向上，探索科研乐趣，不轻易放弃任何一个学生。您敏捷的思维逻辑、深刻的科研见解、深厚的学术功底、严谨的

科研态度令我铭记于心。成为我不断学习，不断奋斗的目标榜样。感谢您带我度过这充实而难忘的三年时光。衷心祝愿庞老师阖家幸福，事业再上一层楼！

感谢实验室的许倩倩老师、王树徽老师、苏荔老师、李国荣老师、齐洪刚老师、李亮老师、张维刚老师和吴益灵师姐、杨智勇师兄、卓君宝师兄、吴哲师兄在学习中给我的帮助。感谢这三年唯一的室友和伙伴廖昌粟同学以及戚兆波、徐凯、辛永健、刘雪静、胡玲、郭双双等同学的陪伴，使得这三年的时光也有许多欢声笑语。在此祝愿所有的老师工作顺利，所有的同学们学业有成！

感谢未来的她，是你让我在迷茫困惑时有了坚持下来的动力！

最后，向百忙之中抽出宝贵时间评审本论文的专家和学者表示感谢！