复杂文本的主题挖掘与可视分析

（申请清华大学工学博士学位论文）

培 养 单 位：高 等 研 究 院

学　　科：计 算 机 科 学 与 技 术

研　究　生：王　希　廷

指 导 教 师：郭　百　宁　教　授

联合导师：温　江　涛　教　授

二〇一六年十月

Topic Mining and Visual Topic Analysis of Rich Text Corpora

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Wang Xiting

Dissertation Supervisor : Professor Guo Baining

Cooperate Supervisor : Professor Wen Jiangtao

October, 2016

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

主题，即文本中谈论的主要内容，通常表示为词的概率分布。主题分析在市场分析和舆情分析等方面有着重要的作用，是文本分析中十分重要的研究内容。随着计算机网络和信息技术的广泛应用，文本数据日趋复杂，给主题分析带来诸多挑战。文本的复杂性主要体现在两个方面。第一是文本数据源多。我们经常接触的文本源就包括新闻、微博、博客等。这些文本源的侧重点不同，用语习惯差异也大，综合分析多个文本源的主题并不容易。第二是文本内容随时间动态变化。因此，主题内容也动态变化，不同时间点的主题之间还相互关联。分析这些复杂的变化和关联并不容易。本论文主要围绕文本的这两点复杂性展开了研究。

具体来说，本论文将复杂文本分为单源动态文本、多源静态文本以及多源动态文本三类。针对这三类文本中主题分析的难点，本论文提出了一系列主题挖掘和可视分析方法，帮助用户快速、有效地分析复杂文本中丰富的主题信息。

单源动态文本方面，本论文提出了动态主题树的构建及可视化方法，帮助用户分析大量主题及其关联的动态变化。为了有效构建动态主题树，本论文提出了贝叶斯在线滤波框架。该框架一方面保证主题树正确反映文本中的主题分布，另一方面确保主题树的动态变化正确反映文本内容的连贯性。

多源静态文本方面，本论文提出了主题全景图的挖掘与可视交互方法，帮助用户综合分析多个文本源中的主题。主题全景图由不同文本源的主题图拼接而成，既包含多个文本源共有的主题，又包含单个文本源独有的主题。为了准确生成主题全景图，本论文提出了一致的图匹配算法，并且允许用户通过交互对图匹配结果进行增量式修改。另外，本论文设计了基于密度的图可视化方法帮助用户分析含有大量主题的全景图。

多源动态文本方面，本论文提出了相关主题之间领先-滞后关系的挖掘与可视化方法，帮助用户分析不同时间点、不同文本源的相关主题之间复杂的关联。为了准确提取动态领先-滞后关系，本论文提出了基于随机游走的相关模型与张量分解技术。为了帮助用户分析长的时间跨度上大量的动态领先-滞后关系，本论文设计了气泡树、基于相关聚类的流向图来减少视觉混乱和歧义。

关键词：多源文本；动态文本；可视分析；主题分析；主题树

Abstract

Topics, each of which is characterized by a distribution of words, summarize the main ideas of a textual document. Topic analysis is important in many applications such as market analysis and public opinion analysis. However, topic analysis is challenging due to two reasons. First, there are many textual sources (e.g., news, blogs, micro-blogs) on the Internet. Different textual sources have their own unique topics and contain texts with different language usages. Thus it is difficult to use a unified method to learn topics that fit each source well. Second, document content often evolves over time. As a result, the content of a topic and topic relations also evolve over time. Tracking the evolution of topics and their relations from huge amount of textual data is technically demanding.

To tackle these challenges, we propose a series of mining and visual analytics methods to help users better analyze topics discussed in time-varying and multi-source textual data.

For *time-varying textual data from a single source*, we develop a method that learns evolutionary multi-branch topic trees and their evolution patterns over time. To effectively model evolutionary multi-branch trees, we propose a Bayesian online filtering framework that jointly optimizes the fitness of each tree and the smoothness between adjacent trees.

For *static textual data from multiple sources*, we develop an approach to analyzing a full picture of topics discussed in multiple sources. First, we model each textual source as a topic graph. We then match these graphs together with a consistent graph matching method. Next, a level-of-detail visualization is designed to enhance users' ability to understand and analyze the matched graph. We also allow users to interactively modify the matched graph based on their information needs.

For *time-varying textual data from multiple sources*, we propose an approach to analyzing lead-lag relationships between correlated topics from multiple sources. To accurately mine lead-lag relationships, we develop a random-walk-based correlation model and combine it with tensor decomposition. To convey complex lead-lag relationships, we design a visualization that combines the strengths of a bubble tree, a correlated-clustering-based flow map, and a focus+context timeline.

Key words: multi-source textual data; time-varying textual data; visual analytics; topic analysis; topic tree

目 录

主要符号对照表

AIC	赤池信息准则（Akaike Information Criterion）
BRT	贝叶斯多分枝树（Bayesian Rose Tree）
CTM	相关主题模型（Correlated Topic Model）
DAG	有向无环图（Directed Acyclic Graph）
DTM	动态主题模型（Dynamic Topic Model）
HDP	层次化狄利克雷过程（Hierarchical Dirichlet Processes）
LOD	多层次细节（Levels of Detail）
MRF	马尔科夫随机场（Markov Random Field）
NMF	非负矩阵分解（Non-negative Matrix Factorization）
NMI	归一化互信息（Normalized Mutual Information）
$\mathcal{D}$, $\mathcal{D}^t$	（t 时刻的）文档集合
$\mathcal{R}$	实数集合
$\|\mathcal{S}\|$	某集合 $\mathcal{S}$ 中的元素个数
T, T^t	（t 时刻的）主题树
T_i, T_i^t	（t 时刻的）子主题树
$\mathbf{x}_i$, $\mathbf{x}_i^t$	（t 时刻的）文档集合中的第 i 个文档

第 1 章　引言

主题，即文本中谈论的主要内容，通常由词的概率分布来表示。例如，一个与埃博拉病毒相关的主题可以表示为：（埃博拉，0.4），（病毒，0.2），（致死，0.2），（传播，0.1），（体液，0.1）。这里，括号中的数字代表了相应的词在主题中出现的概率。主题分析可以帮助用户快速了解大量文本中的主要内容，在市场分析和舆情分析等方面有着重要的作用。例如，它可以帮助公司管理者从大量用户反馈中分析用户主要意见，从而调整经营策略；帮助政治家从大量选民意见中分析舆论导向产生的原因，从而合理应对公共关系危机；帮助政府官员从大量社交网络文本数据中理解民众对危机事件（例如埃博拉）的看法，从而引导舆情向更理性、健康的方向发展。因此，主题分析是文本分析中十分重要的研究内容。

主题分析主要有两个研究目标。第一个目标是有效性高。有效性包含主题信息准确性（G1）以及主题信息丰富性（G2）。由于主题是对文本内容的高度总结，主题信息提取不准确，可能造成用户对文本内容理解有误、理解不全面或者是难以理解等问题。因此，准确提取主题信息是一个重要的研究目标。主题信息丰富主要指可以分析主题随时间的变化与主题之间丰富的关联。一个著名的例子是主题随时间的合并与分裂[1]。如图 1.1所示，主题的分裂即一个主题分解为多个主题，主题的合并即多个主题融合为一个主题。分析主题的分裂、合并有助于用户找到关键事件以及它们发生的原因，是主题分析中的重要研究内容。主题分析的第二个研究目标是可扩展性好，即支持对大量主题进行分析（G3）。网络上文本数据量日益增加，主题数目也日趋增多。一个用户关心的事件，例如埃博拉，往往包含成百上千的主题。只有支持对大量主题的分析，才能帮助用户全面地了解这个事件。

随着计算机网络和信息技术的广泛应用，文本数据日趋复杂，给有效分析大量

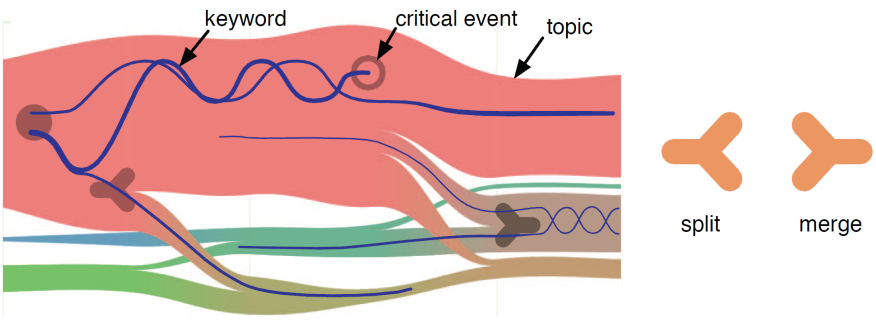


图 1.1　主题的分裂（Split）与合并（Merge）。图中，x 轴方向表示时间，每个条带代表一个主题，条带的宽度随时间变化，代表主题热度随时间变化。（图片引自 [1]）

主题带来了诸多挑战。本论文着眼于复杂文本中的主题分析，对其面临的挑战以及主要研究思路进行了探讨，并针对这些挑战展开工作，提出了我们的解决方案。

1.1　复杂文本主题分析面临的挑战

文本的复杂性主要表现在两个方面。第一，文本的数据源多。一个大事件，例如埃博拉，可能在新闻、博客、微博、维基百科、邮件、论文等多个文本源中都被讨论。目前因特网上的文本源数量已经非常巨大，仅数据提供商 Spinn3r 就存储了来自 1.4 亿个文本源的文本数据。第二，文本的内容随时间动态变化。随着一个事件不断发展，与它相关的文本内容也在不断动态变化。例如，在埃博拉的爆发期、发展期以及回落期，相关文本中的主题内容在不断变化。在爆发期，与死亡人数、埃博拉导致的混乱相关的主题占据主导地位。在发展期，由埃博拉导致的混乱渐渐得到控制，人们的关注点转移到了政府采取的措施。与此同时，死亡人数仍然是受到普遍关注的主题。在回落期，死亡人数得到控制，反弹案例相关主题开始出现。

根据文本的这两点复杂性，我们可以把复杂文本分为单源动态文本、多源静态文本和多源动态文本。如图 1.2所示，这三类文本的复杂度越来越高。下面，我们针对这三类文本的主题分析中存在的主要难点与挑战进行探讨。

单源动态文本挑战：分析大量主题的分裂与合并关系。 如何分析大量主题的分裂与合并关系，从而帮助用户更好地分析关键事件及产生的原因，是单源动态文本方面尚未解决的问题。Cui 等人的方法[1] 可以分析分裂、合并关系，主题信息丰富性（G2）较好。但是他们的方法难以处理数量很多的主题，在处理大量主题（G3）方面有待改进。Dou 等人的方法[4] 可以分析大量主题，方法可拓展性（G3）好。但是他们的方法难以分析主题的分裂、合并关系，在主题丰富性方面有不足之处（G2）。

要分析大量主题的分裂与合并关系，我们需要利用多分枝主题树有效组织主题，并且分析多分枝主题树随时间的动态变化规律。由于多分枝树的结构较为复杂，树上的算法复杂度往往比较高。快速地构建动态多分枝主题树，同时保证方法的有效性是一个具有挑战性的问题。

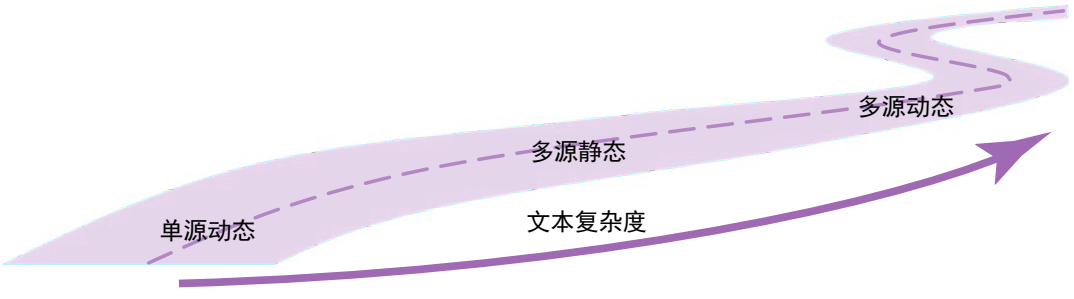


图 1.2　复杂文本：单源动态文本、多源静态文本与多源动态文本。

多源静态文本挑战：准确分析文本源的大量共有主题与独有主题。 多源静态文本的主题分析主要难点在于准确分析文本源的共有主题与独有主题（G1，G2）。由于文本源很多，一个事件相关的主题很可能散落在多个文本源中。不同文本源侧重点有所不同，谈论到的主题也有所区别。以埃博拉为例，新闻中可能主要侧重于事实的报导，博客中可能主要侧重于被埃博拉感染的志愿者的个人介绍，微博中可能会存在一些谣言，例如盐水可以防治埃博拉。如果只看单独的文本源，无法对埃博拉事件中涉及的主题产生完整的了解。只有把多个文本源综合分析，了解文本源的共有主题与独有主题（G2），才能对事件的全貌产生完整的了解，进行更合理地决策。

然而，准确分析文本源的共有主题与独有主题并不容易。不同文本源的文本长度、用语习惯不同，难以用统一的模型综合考虑。例如，新闻中往往是比较规范的长文本，微博中常常是口语化的短文本。如果用同样的模型进行考虑，生成结果的准确性不够令人满意[6]。为了产生准确性较高的结果（G1），我们需要将不同文本源用最合适的模型和参数学习出主题，然后将不同文本源的主题进行匹配。因为相同的主题在不同文本源中可能有差异，因此匹配过程需要允许主题内容有一定的误差（Error-Tolerant）。另外，不同用户对一个事件中不同文本源的共有主题和独有主题理解可能不同，我们还需要允许用户对匹配结果进行修改，从而进一步提高准确性。如何进行允许误差的匹配，直观展现大量主题的匹配结果，并且允许用户对匹配结果进行增量式修改，是目前仍待解决的问题。

多源动态文本挑战：准确分析不同文本源中大量相关主题的领先-滞后关系。 近期，一些研究人员开始对多源动态文本的主题分析进行研究。一个代表性的工作是 2014 年 Liu 等人[3] 提出的对多源文本中领先-滞后关系进行分析的方法。Liu 等人提出，同一个主题上，不同文本源中可能存在领先-滞后关系。例如，本拉登死亡相关的主题在推特（Twitter）[7] 中首先被提到，然后才开始在新闻中大规模报导。因此，在本拉登这个主题上推特领先于新闻。分析领先-滞后关系非常重要，例如，它可以帮助公司管理者了解哪个文本源更有影响力，从而优化公关策略。

Liu 等人的方法使得主题信息丰富性（G2）有所提高，但是在主题分析准确性方面（G1）仍然有不足之处。Liu 等人的算法只利用文本内容进行领先-滞后关系的计算，忽略了时间序列相关性以及文本丰富的元数据（例如文章的引用关系、作者之间的好友关系等），因此准确性不够高。另外，该方法可以分析同一个主题上不同文本源之间的领先-滞后关系，但是忽略了不同的相关主题之间的领先-滞后关系，因此在主题信息丰富性（G2）方面仍有所不足。例如，经济和政治尽管是不同的主题，但是它们之间可以互相影响，也存在领先-滞后的关系。如何综合考虑文本内容、文本中词的时间序列的相关性以及文本中丰富的元数据准确提取多源

动态文本中大量相关主题的领先-滞后关系，是多源动态文本分析中的一大难点。

1.2　复杂文本中主题的分析思路

要分析复杂文本中的主题，传统思路主要是主题挖掘与文本可视化。下面，我们简要介绍这两种思路，并且对它们的优缺点进行探究。基于这些探究，我们提出了本论文采用的主要分析思路：主题可视分析。

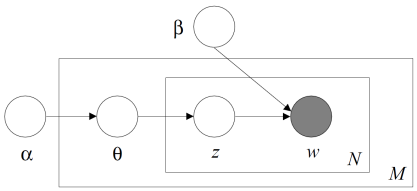
主题挖掘。主题挖掘利用词的共现（Co-occurrence）关系或者文本内容相似性等信息自动从文本中提取主题。图 1.3(a)展示了一个经典的文本挖掘算法 LDA（Latent Dirichlet Allocation）[8] 的图模型。图中，深色圆代表观测变量，白色圆代表隐变量（Latent Variables），箭头代表随机变量之间存在依赖关系，方框相当于统计意义上的 for 循环。这里，M 代表文档数量，N 代表文档中词的个数，α 是给定的 k 维向量，k 代表需要学习的主题个数。LDA 主题模型认为，每篇文档的主题分布 θ 服从狄利克雷分布 $Dir(\alpha)$。文档中第 n 个词 w_n 与它对应的主题 z_n，以及主题对应的词的概率分布 β 都有关，表现为 $p(w_n|z_n, \beta)$ 是一个多项条件概率分布。给定一组文档，LDA 通过最大化这组文档出现的后验概率来学习模型中对应的参数，从而获得主题词的概率分布。除了 LDA 以外，常用的主题挖掘模型还包括动态主题模型 DTM（Dynamic Topic Models）[9]，考虑主题之间相关性的模型 CTM（Correlated Topic Models）[10]，以及学习多分枝层次主题的模型贝叶斯多分枝树[5] 等。

主题挖掘的优点是可以较准确地自动提取主题，节省人力。缺点主要有三个。

首先，主题挖掘生成的结果不够直观，用户难以理解。每个主题对应着一个词的概率分布，每篇文档对应着一个主题的概率分布。要让普通用户理解挖掘结果，需要有更加直观的展现形式。LDA 的作者 Blei 等人利用如图 1.3(b)所示的列表与如图 1.3(c)所示的对词着色的方式分别展现了四个主题中词的分布和文档中主题的分布。这种方法尽管可以帮助用户从一定程度上了解主题信息，但是当主题、文档个数变多或者主题信息变丰富（例如，主题信息包含主题动态变化或者主题之间关联信息）时，主题挖掘结果将更难被普通用户理解。

其次，主题挖掘生成结果无法进行用户定制，难以满足不同用户的不同需求。尽管主题挖掘结果较为准确，但是并不是完美的。一方面，挖掘结果中可能含有错漏，另外一方面，不同用户的需求可能不尽相同，他们所需要的主题挖掘结果也不完全相同。要纠正挖掘结果中的错漏之处，生成更满足用户需求的结果，都需要用户的干预。但是因为缺乏交互机制，主题模型的结果无法有效利用用户的知识，造成了一定的局限。

最后，主题挖掘对于目的模糊的分析任务的支持较弱。现实生活中，分析任务



(a) LDA 图模型

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT

(b) 挖掘结果一：主题内容

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

(c) 挖掘结果二：文档中每个词对应的主题

图 1.3　LDA（Latent Dirichlet Allocation）主题挖掘模型及结果。（图片引自 [8]）

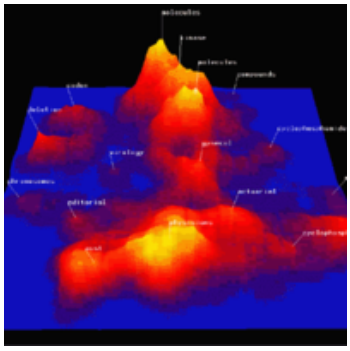


图 1.4　James 等人[11] 利用文本可视化的方法进行文本主题分析的结果。（图片引自 http://textvis.lnu.se/）

往往是比较模糊的。例如，公司管理者的任务可能是通过分析大量文本，了解如何提高本公司产品的竞争力。学者的任务可能是通过分析大量论文，推测最适合自己的研究方向。这类任务的答案难以直接通过文本挖掘结果得到，需要人对挖掘结果进行不断地探索、理解、整理才可能得到。在这个过程中，用户往往需要从多个角度了解挖掘结果，通过当前关注的内容找到相似的其他内容，并且与他人分享讨论自己的心得。这类需求很难被单纯的主题挖掘方法所满足。

文本可视化。 文本可视化的方法也可以对文本中的主题进行分析。这类方法的主要思想是通过对文档或者词的合理布局体现出文档的主题。例如，James 等人[11] 把相似的文档布局在二维平面的相近位置。如图 1.4所示，这样，同一个主题的文档自然地聚集在一起，形成文档类，用户可以直观地看到不同的主题以及主题中

表 1.1　对比三种对复杂文本主题进行分析的思路。文本可视分析综合了主题挖掘和文本可视化的优点。

分析思路	节省人力	结果易于理解	可以用户定制	支持目的模糊的分析任务
主题挖掘	是	否	否	否
文本可视化	否	是	是	是
主题可视分析	是	是	是	是

含有文档的多少。通过交互，用户还可以研究具体的主题或者文档，或者找到和当前感兴趣主题相关的其他主题。

文本可视化的方法生成结果非常直观，容易被用户理解。通过交互，用户可以对视图进行修改，生成更符合自己需求的结果。另外，用户还可以利用交互从多个角度了解挖掘结果，通过探索不断明确自己的分析任务，因此文本可视化可以较好地分析目的较为模糊的任务。但是，文本可视化不直接提取主题，很多时候具体主题的分析还是需要依赖用户，不够节省人力。当需要分析的主题信息较为复杂时（例如，需要分析随时间动态变化的主题时），用户的分析负担进一步增大。因此，单纯的文本可视化也难以满足现实生活中对复杂文本的分析需求。

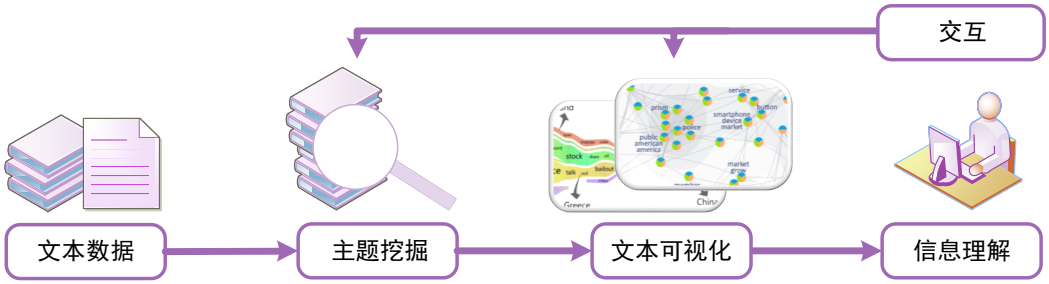


图 1.5　主题可视分析流程图。

主题可视分析。为了结合两种方法的优点，我们利用可视分析技术把主题挖掘和可视化结合在一起。如图 1.5所示，可视分析将数据挖掘的结果直观地展现给用户，因此易于理解。用户还可以通过交互修改文本挖掘模型，获得用户定制的挖掘结果，并且进一步提高准确性，也支持对目的模糊的任务进行分析。表 1.1总结了主题挖掘、文本可视化与文本可视分析三种思路的优缺点。可以看到，可视分析把机器擅长的事情交给机器，把人擅长的事情交给人，综合了文本挖掘和文本可视化两者的优点。

1.3　论文的主要工作

本论文的主要工作是开发数据挖掘与可视分析技术，让用户准确、高效地分析单源动态文本、多源静态文本以及多源动态文本中的丰富的主题信息。图 1.6简

要总结了本论文的主要工作。下面，分别就单源动态文本、多源静态文本以及多源动态文本介绍我们的主要贡献。

1.3.1　单源动态文本的主题挖掘与可视分析

单源动态文本方面，本论文提出了一种快速、有效地对大量主题的分裂、合并关系进行分析的数据挖掘方法，并用现有的可视化技术对挖掘结果进行了直观展现。该方法利用多分枝主题树组织主题，支持对大量主题的分析。通过分析主题树随时间的动态变化，该方法可以分析大量主题的分裂、合并关系。

我们面临的技术难点有两个。首先，我们需要同时保证动态多分枝主题树的拟合度（Fitness）与平滑度（Smoothness）。拟合度高即各个时刻的主题树都能正确反应文本中的主题分布。平滑度高即当文本内容变化不剧烈时，主题树之间相似度较高。平滑度高保证了动态主题树能够正确反应文本内容的连贯性。其中确保多分枝树之间的平滑度难度很大，现有的方法尽管能加平滑约束，但是平滑效果不理想。另外，我们还需要设计统一的模型，来综合优化树的拟合度与平滑性，这也是一个尚未解决的问题。第二个难点是要保证算法的效率。这是因为树上的算法一般复杂度比较高，难以处理大量的文本。

为了准确地挖掘动态多分枝主题树，本论文设计了一个贝叶斯在线滤波（Bayesian Online Filtering）框架为变化的多分枝树建模。这个框架使得我们可以同时对拟合度和平滑度进行优化。为了解决现有方法平滑效果不理想的问题，我们引入了三元组约束（Triple）与扇形约束（Fan）。为了提高算法的效率，我们建立了树的索引，将算法时间复杂度从 $\Omega(n^3)$ 降低到了 $O(n\log(n))$。最后，我们利用 TextFlow[1] 对主题挖掘结果进行可视化，使得结果直观易于理解。

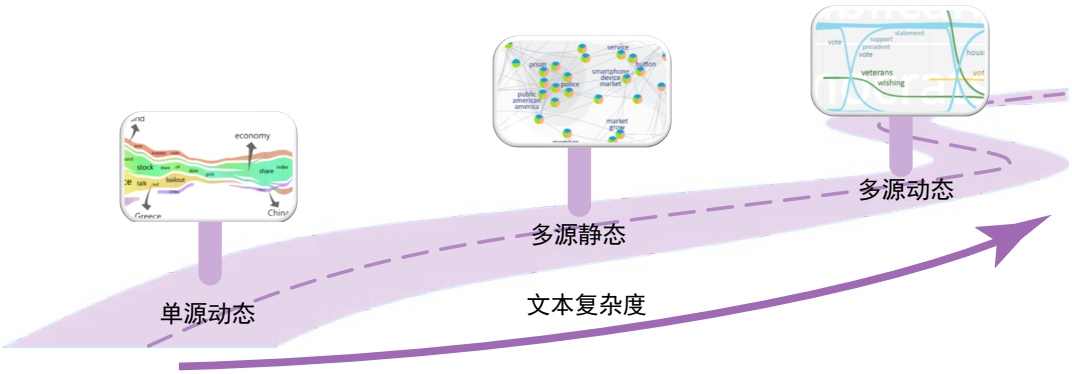


图 1.6　本论文的主要工作：开发主题挖掘与可视分析技术，让用户准确高效地分析复杂文本中丰富的主题信息。

1.3.2　多源静态文本的主题挖掘与可视分析

多源静态文本方面，本论文提出了对多个文本源中大量共有与独有主题进行挖掘和可视分析的方法。我们的思路是从每个文本源提取一张主题图，并将不同文本源的主题图拼接在一起，生成一张主题全景图。主题图中，每个点代表一个主题，每条边代表主题之间的关联。将不同文本源的主题拼接在一起形成的主题全景图既包含了多个文本源的共有主题，也包含了单个文本源的独有主题。用户可以通过分析主题全景图了解事件的全貌。

要实现上面的想法，主要有以下难点。第一个难点是准确提取主题全景图。主题全景图的提取需要用到图匹配（Graph Matching）技术。目前，图匹配技术主要用于两个图的匹配，如果直接用这些算法对图两两进行匹配，很容易产生不一致的、互相矛盾的匹配结果。第二个难点是如何产生用户定制的图匹配结果。图匹配算法往往不是完美的，可能存在错漏。另外，不同的用户所需要的图匹配结果也往往不太一样。这需要我们的算法支持图匹配结果的实时修改。如何快速、有效地根据用户知识更新图匹配结果也是一个难点。第三个难点是要有效展示主题全景图。主题全景图中往往含有几百个主题。如何有效展示大量的共有主题和独有主题，从而让用户快速了解全景图，是一个值得探究的问题。

为了准确提取主题全景图，我们设计一致的图匹配算法。我们提出元图（Meta-graph）的概念，指出任何一致的匹配结果都可以表示为一个元图。我们的算法首先生成元图，然后迭代地对元图进行修改，可以保证在对三个或以上的图进行匹配时，也能产生较好的一致的结果。另外，我们提出基于度量学习（Metric Learning）与特征选择（Feature Selection）的增量式修改图匹配结果的方法。我们的方法可以有效地利用用户知识更新图匹配结果，生成更符合用户需求的主题全景图。为了有效展示大量的共有主题与独有主题，我们设计了基于 LOD（Levels of Details）的可视化方法。该方法结合径向冰柱树（Radial Icicle Plot）与基于密度的图可视化，可以有效展示大量主题，同时保证用户可以自由地对主题全景图进行放大（Drill In）、缩小（Drill out）。另外，我们开发了基于 Voronoi 剖分的布局算法，可以有效区分共有主题和独有主题。

1.3.3　多源动态文本的主题挖掘与可视分析

多源动态文本方面，本论文提出了对多个文本源中大量相关主题的领先-滞后关系进行挖掘和可视分析的方法。分析文本源中大量相关主题的领先-滞后关系是多源动态文本分析的一大难点与挑战。要解决这个问题，需要克服一系列的技术难点。首先，我们需要准确提取多个文本源相关主题的领先-滞后关系。现有方法考

虑的数据类型比较单一，准确性不够令人满意。另外，目前缺乏统一的模型考虑两个以上的文本源的相关主题间的领先-滞后关系。现有方法或者只考虑同一个主题在不同文本源的领先-滞后关系，忽略了不同但是相关的主题之间的领先-滞后，或者只能分析两个文本源相关主题间的领先-滞后。第二个技术难点是有效展示多个文本源相关主题的领先-滞后关系。领先-滞后关系涉及到多个文本源、多个主题、多个时间点，很容易造成视觉混乱（Visual Clutter）及歧义。如何减少视觉混乱和歧义，是一个有待研究的问题。

为了准确提取多个文本源相关主题的领先-滞后关系，我们开发了基于随机游走的挖掘模型。该模型利用随机游走相关模型（Random-walk-based Correlation Model）综合考虑文本内容、时间序列相关性以及文本元数据，提高了多源动态主题提取与领先-滞后关系提取的准确性。另外，该模型利用张量统一考虑多个文本源，可以计算三个及以上文本源相关主题之间的领先-滞后关系。为了减少领先-滞后关系展示时产生的视觉混乱，我们设计了基于 Voronoi 树图的气泡树、基于相关聚类的流向图以及焦点加上下文的时间轴，使得用户可以快速有效地分析涉及多个文本源、多个主题、多个时间点的领先-滞后关系。

1.4　论文概览

本论文后续章节组织总结如下。第 2 章介绍复杂文本的主题挖掘与可视分析的研究现状，对现有工作的优缺点进行了分析探讨。第 3 章、第 4 章、第 5 章分别介绍我们在单源动态、多源静态以及多源动态文本方面的工作。第 6 章对我们的工作进行了总结，并对未来研究方向进行了讨论。

第 2 章　相关工作

复杂文本的主题分析在文本挖掘、文本可视化与文本可视分析领域都是被广泛研究的问题。下面，本论文分别介绍单源动态文本、多源静态文本以及多源动态文本方面的研究现状，并从方法的有效性（即主题提取准确性与主题信息丰富性）和可拓展性出发，探讨方法的有优点与局限性。

2.1　单源动态文本主题分析的研究现状

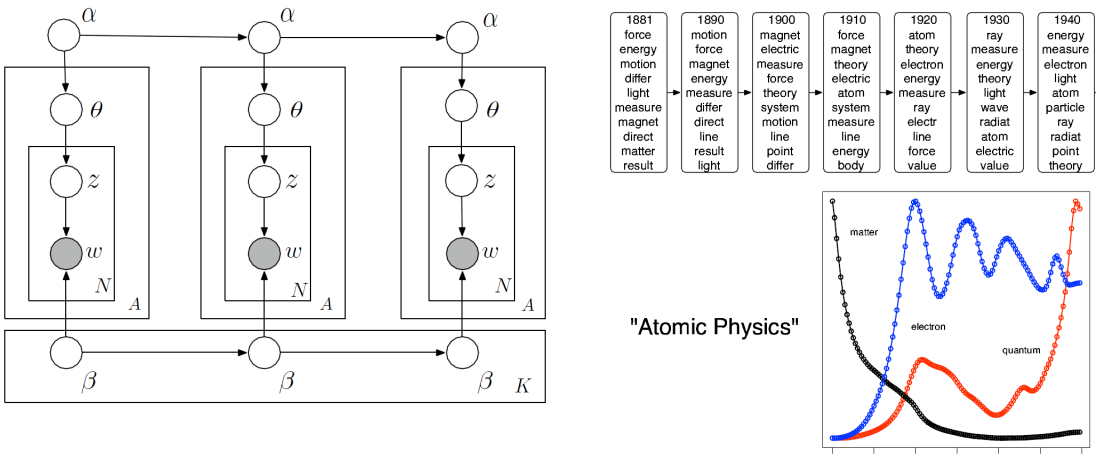
单源动态文本的主题分析方法可以按照提取的是非层次化主题还是层次化主题分为两类。非层次化主题挖掘模型提取的主题之间没有从属关系，都在同一层级上。这类方法在分析丰富的主题信息方面有优势，但是不适合分析大量的主题，可拓展性较弱。层次化主题挖掘模型利用树的结构组织主题，可以分析大量主题，但是在主题信息的丰富性上尚有所欠缺。表 2.1总结了这两类方法的优缺点。下面对这两类方法的工作进行具体介绍与探讨。

表 2.1　单源动态文本主题分析的研究现状。

	有效性		可拓展性
	主题信息准确性	主题信息丰富性	分析大量主题
基于非层次化主题挖掘模型	较好	较好	较差
基于层次化主题挖掘模型	较好	较差	较好

2.1.1　基于非层次化主题挖掘模型的方法

数据挖掘方法。 挖掘随时间动态变化的主题的方法最早由 Blei 等人[9] 提出。2006 年，Blei 等人提出 DTM（Dynamic Topic Model）主题模型。DTM 在三个时间点上的图模型如图 2.1(a)所示。DTM 在单个时间点的图模型和 LDA 的图模型（图 1.3(a)）类似，主要区别在于它的两个主要参数 α 与 β 与上一时刻的参数相关。这里，α 是控制文档中主题分布的参数，β 是所有主题对应的词的分布。DTM 模型规定，某时刻的 α 与 β 与上一时刻的 α 与 β 相差高斯噪音。通过保证在文本内容没有发生剧烈变化时 α 和 β 变化较小，就可以得到随时间平滑变化的主题。DTM 的部分挖掘结果如图 2.1(b)所示。图中展示的是《科学》杂志中，与“原子物理”相关的主题内容随时间的动态变化。图的上半部分显示的是该主题在不同时间点的前十个



(a) DTM 图模型　　(b) DTM 挖掘结果

图 2.1　DTM（Dynamic Topic Models）主题挖掘模型及结果。（图片引自 [9]）

关键词，下半分显示的是该主题中三个关键词的后验频率随时间的变化。可以看出，动态主题挖掘可以较好分析文本中主题内容的变化趋势。

DTM 可以较为准确地提取主题内容的动态变化，但是在主题信息丰富性方面还有所欠缺。为了解决这个问题，Gao 等人[13] 利用增量式层次化狄利克雷过程提取了动态主题的出生、死亡、分裂以及合并关系。Ahmed 等人[14] 通过将周期性中餐馆过程（Recurrent Chinese Restaurant Process）与 LDA 结合，提出了一个统一的框架。该框架可以同时提取一个故事对应的人物、时间、地点与主题。以上方法主要利用了文本内容信息，Wang 等人[15] 综合文本内容信息与文档之间的引用关系，提出了 Citation-LDA 模型。该算法不仅可以提取主题内容、强度、重要性随时间的动态变化，还可以提取主题之间的依赖关系，以及主题中最关键的文档。

数据挖掘的方法能较为准确地提取主题，主题信息也较为丰富。但是因为主题信息较为复杂，用户难以理解，因此难以分析大量的主题。

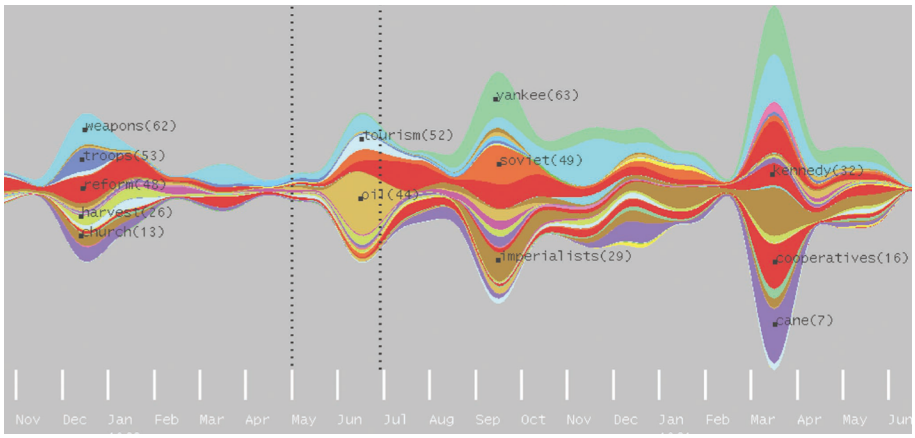


图 2.2　ThemeRiver 利用河流作为视觉隐喻表现动态变化的主题。图中，x 轴代表时间，每个条带代表一个主题，条带的宽度代表主题在特定时间的热度。（图片引自 [12]）

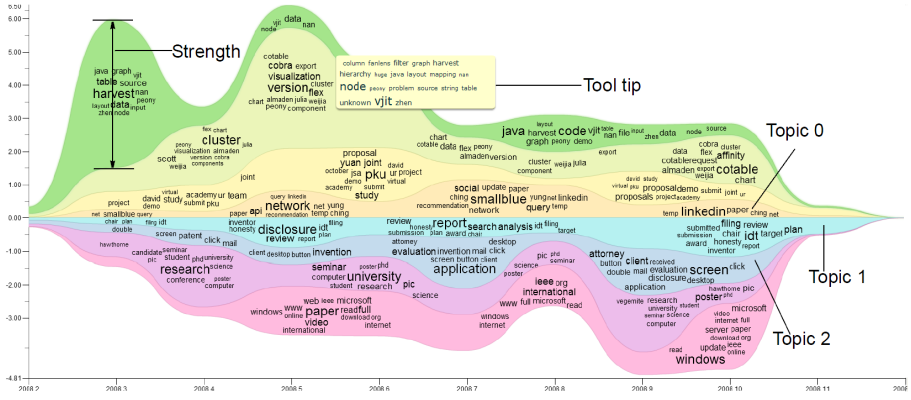


图 2.3　TIARA 可视分析系统界面。图中，x 轴代表时间，每个条带代表一个主题，条带上的关键词展示了主题的内容，条带的宽度代表主题在特定时间的热度。（图片引自 [16]）

可视分析方法。 研究者也提出了一些可视分析的方法帮助人们更好地理解复杂的主题挖掘结果。ThemeRiver[12] 是最早提出用河流的视觉隐喻（Visual Metaphor）来展示随时间动态变化的主题的可视分析系统。ThemeRiver 如图 2.2 所示。图中，x 轴代表时间，不同颜色的条带代表不同的主题。条带的宽度随 x 轴变化，表示主题热度随时间不断变化。TIARA[16–18] 将叠式图（Stacked Graph）与 LDA 紧密结合，将主题对应的关键词合理布局在主题对应的条带上（图 2.3），可以展示比 ThemeRiver 更丰富的主题信息。Visual Backchannel[19] 除了用河流的形式展示动态变化的主题，还用螺线（Spiral）展现了主题对应的人，用图云（Image Cloud）展现了相关图片。ParallelTopics[20] 利用 ThemeRiver 展现主题随时间的动态变化，另外通过平行坐标（Parallel Coordinates）展现文档中主题的概率分布。

以上方法可以帮助用户直观、有效地分析主题内容以及主题强度随时间的变化，但是无法分析主题之间丰富的关联及其随时间的动态变化。2011 年，Cui 等人设计了 TextFlow[1]，利用桑基图（Sankey Diagram）表现主题的动态分裂、合并关系。TextFlow 还提取了关键事件以及关键词之间的相关性。如图 1.1所示，主题的分裂、合并用条带的分裂、合并表示，关键事件用特定符号（Glyph）表示，关键词用关键词线（Keyword Thread，图中蓝色曲线）来表示。通过观察 TextFlow，用户不仅可以分析主题的分裂、合并，还可以找到关键事件，分析关键事件产生的原因。受到 TextFlow 中视觉隐喻的启发，Gad 等人提出了 ThemeDelta[21]。ThemeDelta 可以分析多个关键词是怎么合并到一个主题，或者分裂到不同的主题。另外，它还可以分析主题随时间动态变化的趋势、词的聚类结果以及主题产生变化的关键点。

随着以推特为代表的社交网络的兴起，研究社交网络上主题的动态变化的工作也开始出现。2014 年，Xu 等人[22] 指出社交网络上的主题之间存在竞争关系，并且设计了可视交互界面，帮助用户理解主题之间的竞争关系如何随时间动态变化。2015 年，Sun 等人[23] 提出，社交网络上的主题之间不仅存在竞争关系，同时还存

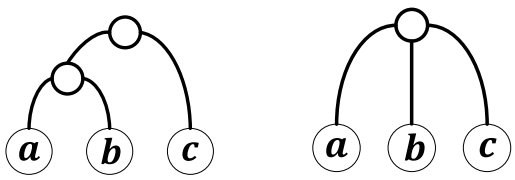
在着合作关系，这两种关系随时间的动态变化主要是由主题领袖造成的。为了帮助用户同时分析主题之间的合作与竞争关系，Sun 等人提出了可以同时提取主题间竞争、合作关系的数据挖掘模型。该模型还可以同时提取影响这两种关系的意见领袖。此外，Sun 等人还开发了 EvoRiver 可视分析系统，并且通过案例分析验证了分析主题间竞争合作关系的应用价值。

相比于数据挖掘的方法，基于可视分析技术的方法能够更加直观地展现主题信息，可扩展性有所提升。但是因为上述方法采用的是非层次化主题挖掘方法，不能有效地组织大量主题，所以最多只能分析几十个主题的动态变化，不能很好地满足对大量主题进行分析的需求。

2.1.2　基于层次化主题挖掘模型的方法

数据挖掘方法。 数据挖掘算法中与挖掘动态变化的层次化主题（主题树）相关的工作是带约束的层次化聚类。如果把 $t-1$ 时刻的主题树分解为一系列约束，我们可以通过带约束的层次化聚类建立第 t 时刻的主题树。重复这个过程就可以得到动态变化的主题树。带约束的层次化聚类方法可以按照约束类型分为两类。

第一类方法用的是成对约束（Pairwise Constraints）。成对约束规定一对样本必须在同一个类里（Must-link）或者必须不在同一个类里（Cannot-link）。成对约束最早在非层次化聚类中使用[24,25]。2009 年，Davidson 等人研究了将 Must-link 和 Cannot-link 作为必须要满足的约束（Hard Constraint）加入凝聚式层次聚类方法（Agglomerative Hierarchical Clustering）时算法的复杂度以及何时存在可行解。Davidson 等人指出，当 Cannot-link 是必须要满足的约束时，凝聚式层次聚类可能没有可行解。2011 年，Miyamoto 等人[26] 提出了不将 Cannot-link 作为必须要满足的约束，而是作为软约束（Soft Constraint）的凝聚式层次聚类方法。Miyamoto 等人通过惩罚打破 Cannot-link 的聚类结果，生成尽可能满足约束的结果。在这个方法中，Must-link 既可以作为必须满足的条件，也可以作为软约束。这些方法可以生成比非约束聚类更符合需求的结果。但是因为 Must-link 和 Cannot-link 不包含层



(a) 三元组约束 $ab|c$　　(b) 扇形约束 (abc)

图 2.4　层次化聚类的两种层次约束：(a) 三元组约束，该约束规定样本 a 与样本 b 必须先合并，然后才能与样本 c 进行合并；(b) 扇形约束，假设 $ancestor(a,b)$ 代表样本 a 和 b 最底层的共同祖先，该约束规定 $ancestor(a,b)$、$ancestor(a,c)$ 与 $ancestor(b,c)$ 是同一个节点。

次信息，这些方法无法准确刻画层次化主题的结构。

第二类带约束的层次化聚类方法用的是三元组约束（Triple Constraints）。如图 2.4(a)所示，三元组约束规定两个样本在与特定的第三个样本进行合并之前，必须彼此之间先合并。这种约束包含了层次信息，生成的层次结构优于利用成对约束的层次化聚类方法。现有方法考虑了两种使用三元组约束的方法。第一种方法是基于度量（Metric-based）的方法[27–30]。基于度量的方法通过三元组约束学习样本之间的距离或者相似度，然后在聚类过程中用到学习出来的距离或者相似度。第二种方法是基于实例（Instance-based）的方法[29,31,32]。这种方法在自底向上的聚类过程中保证所有三元组约束都被满足。如果没有办法满足这些约束，将返回聚类失败的结果。尽管利用三元组约束的层次化聚类方法在聚类效果上优于利用成对约束的层次化聚类方法，但是仍然有不足之处。首先，图 2.4(a)所示的三元组约束只能较好地刻画二分枝树的层次信息，但是对多分枝树的刻画能力不足。这是因为多分枝树种包含如图 2.4(b)所示的扇形结构。如何利用扇形约束对多分枝树进行聚类是一个有待研究的问题。其次，要利用带约束的层次聚类方法建立动态主题树，我们还要解决从 $t-1$ 时刻的主题树中提取约束的问题。为了保证提取动态主题树的效率，我们需要保证约束的提取和计算过程效率较高。

可视分析方法。 目前，可视分析方面利用层次化主题对动态单源文本进行分析的工作是 ParallelTopics[20] 与 FluxFlow[33]。这种方法中建立的主题树都是静态、不随时间变化的。尽管它们可以分析大量主题，但是却无法分析主题的动态分裂、合并关系，在主题信息丰富性（G2）方面有所不足。

2.2　多源静态文本主题分析的研究现状

多源静态文本的主题分析可以按照主要思想分为三类。第一类是基于主题模型的方法。这类方法主要通过拓展一种现有的主题模型（例如 LDA），使得主题模型可以同时学习多个文本源的主题。这类方法用同一个模型考虑不同文本源，无法根据不同文本源的不同文本类型单独优化，在提取主题的准确性方面有待提高[6]。第二类方法是基于可视图比较（Visual Graph Comparison）的方法。这类方法先针对每个数据集生成一个主题图。主题图中，每个点代表一个主题，每条边代表主题之间的关联。然后通过可视化的方法让用户比较不同的主题图，从而分析不同文本源主题的差别。这种方法可以分析不同文本源共有主题和独有主题，主题信息丰富性较好。但是，它们往往假设对应的主题是一模一样的，无法处理主题在不同文本源有所差异的情况，在提取主题信息准确性方面有所不足。第三类方法是基于图匹配的算法。这类方法可以考虑主题在不同文本源有所差异的情况。但是，

目前图匹配的算法主要针对两个图的情况，对于三个及三个以上文本源的处理有不足之处。另外，目前基于图匹配的方法主要是数据挖掘的方法，没有可视化界面帮助用户分析大量主题。表 2.2总结了这三类方法的优缺点。下面，我们对这三类方法的工作进行具体介绍与探讨。

表 2.2　多源静态文本主题分析的研究现状。

	有效性		可拓展性
	主题信息准确性	主题信息丰富性	分析大量主题
基于主题模型的方法	一般	较好	一般
基于可视图比较的方法	较差	较好	较好
基于图匹配的方法	一般	较好	一般

2.2.1　基于主题模型的方法

数据挖掘方法。 基于主题模型的多源静态文本分析的早期工作主要是数据挖掘方面的。Teh 等人[34,35] 通过层次化狄利克雷过程（Hierarchical Dirichlet Processes, HDP）对多个文本源进行建模。如图 2.5所示，这类方法中所有文本源有一个共有的基础分布 G_0，每个文本源的分布（G_1 至 G_k）都从 $DP(\alpha_0, G_0)$ 中采样得到。其中，DP 是狄利克雷过程，α_0 是集中参数（Concentration Parameter）。这类方法可以较好地挖掘不同文本源之间的共有主题，但是在独有主题的挖掘方面还有不足。Zhai 等人[36] 提出了比较性文本挖掘（Comparative Text Mining）算法，通过拓展现有的主题模型 pLSI，同时学习文本中的共有主题和独有主题。尽管他们的方法可以学习文本中的共有和独有主题，但是由于 pLSI 本身的局限性，算法无法自然地处理新加入的文档。为了解决这个问题，Paul 等人[37] 提出了在 LDA 主题模型的基础上进行拓展，学习文本中的共有主题和独有主题。除了对共有主题和独有

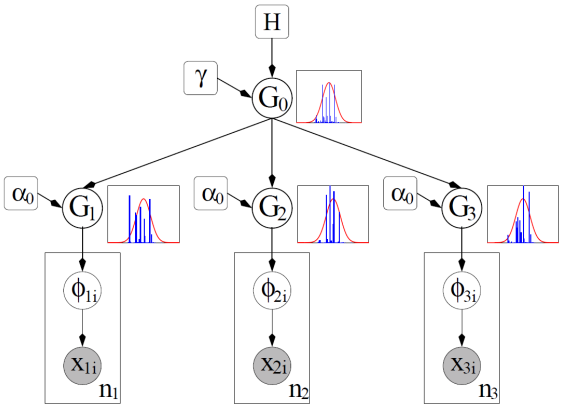


图 2.5　三个文本源时层次化狄利克雷过程（HDP）的图模型。（图片引自 [34]）

主题进行挖掘的方法，还有同时学习多个文本源中主题和主题之间关联度的马尔科夫主题模型[38]，以及利用一个文本集中主题信息优化另一个文本集中主题信息的迁移学习（Transfer Learning）方法 C-LDA[39]。这些方法可以优化原有主题模型的结果，但是并不能直接提取共有与独有主题。

可视分析方法。 近年来，可视分析领域的研究人员也提出了基于主题模型的多源静态文本分析方法。SolarMap[40] 利用密度图展示同一个文本集合与不同文本集合之间的多层面（Multi-facet）上的文档关系。它可以较好地分析含有多层面信息的文档，但是难以直接用于提取主题之间的共有主题与独有主题。Oelke 等人[41] 通过拓展 LDA，提取了不同文本集合的共有主题与独有主题，并利用可视化技术展示了共有和独有主题对应的关键词。这个方法主要问题有两个。第一个是不支持对大量主题的分析。第二个问题是该方法用同样的模型（LDA）处理所有文本集合，当不同集合中文本类型差异较大时，主题提取的准确性不够准确。

以上基于主题模型的方法尽管可以较好地分析文本源的独有和共有主题，但是因为只用一类模型对所有文本源进行统一建模，难以适应不同文本源的文本类型差异较大的情况，在主题提取准确性方面有所欠缺。

2.2.2　基于可视图比较的方法

基于可视图比较的方法利用可视化技术帮助用户分析图之间的共性与差异[42,43]。现有的可视图比较方法可以分为三大类：动画法、并列（Juxtaposition）比较法、叠加（Superposition）比较法。

动画法。 动画法利用动画将一张图平滑地变成另一张图，直观展示图随时间的变化[44–47]。该类方法首先生成一系列随时间动态变化的图，然后对每个时间点的图进行合理布局，使得动画过程中用户容易跟踪感兴趣的节点。布局时需要保证稳定性（Stability）与可读性（Readability）。其中，稳定性指相邻时间点图的布局结果的差异尽可能小，可读性指图的布局结果要正确地反映图的拓扑结构。对图进行了合理的布局以后，该类方法利用动画将一张图渐渐变为下一个时间点的图，帮助用户轻松跟踪图的变化过程。

并列比较法。 并列比较法将两张图在空间或者时间上并列排列[48,49]。经典的并列比较法的例子是 VisLink[50]。如图 2.6所示，VisLink 将每张图的可视化结果展示在独立的二维平面上，然后将对应的节点用线连起来，从而展示不同图之间的关联。2011 年，Bremm 等人[51] 开发了一个可视化工具帮助用户对多棵树进行全局和局部的比较。为了达到这个目的，他们每次只并列展示几棵树。因为每张图都要占用独立的空间，并列比较法难以支持图的张数较多的情况。

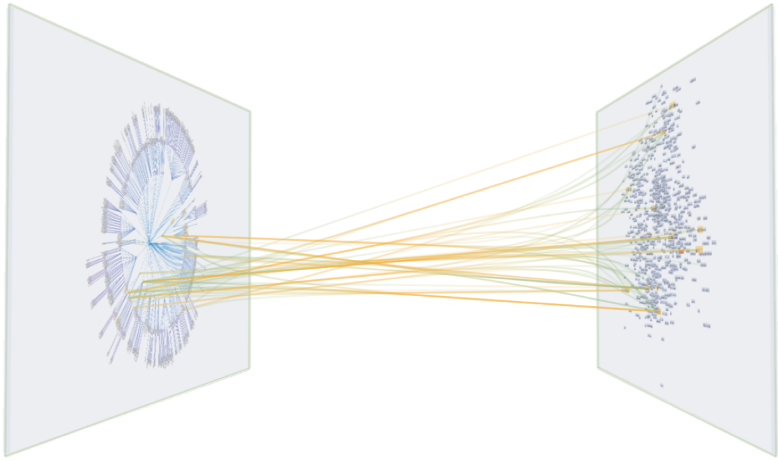


图 2.6　可视图比较方法 VisLink 效果图。

叠加比较法。 叠加比较法将多张图中的相同节点布局在同样的位置。因为相同的点都是重叠的，不占用额外的空间，叠加比较法对于多张图的支持较好。针对不同的比较任务，研究人员开发了不同的叠加比较方法。Alper 等人[44] 将两个矩阵或者点-线图叠加在一起，帮助用户分析比较两张带权图。Vehlow 等人[52] 开发了帮助用户分析网络中重叠社区的可视分析工具。他们利用 LOD 技术，使得用户可以根据需要分析原图或者分析高度聚合的图。

上面这些方法都假设图之间对应的节点是一模一样的。但是，在建立主题全景图时，不同文本源的主题可能并不完全一样，因此上面的方法无法很好地应用在对比主题图中。

2.2.3　基于图匹配的方法

基于图匹配[53] 的算法可以考虑主题在不同文本源有所差异的情况，这类图匹配算法称为允许误差的匹配算法（Error-tolerant）。

数据挖掘方法。 允许误差的匹配算法可以按照针对两个图进行匹配还是可以支持多张图之间的匹配分为两类。大部分允许误差的方法是用于对两个图进行匹配的[54,55]。现有针对两个图进行匹配的方法可以分为基于图编辑距离（Graph Edit Distance）的方法[54,56]、基于人工神经网络的方法[57]、基于松弛标记 (Relaxation Labeling）的方法[58]、基于谱 (Spectral）的方法[59] 以及基于图核（Graph Kernel）的方法[60]。其中，最常用的是基于图编辑距离的方法[54–56,61]。给定两张图，这类方法首先计算一张图转变成另一张图所需的编辑代价。然后通过这个代价估算两张图之间结构的相似性，将两张图相似度较高之处匹配在一起。上面这些方法尽管可以较好地对两张图进行匹配，但是它们并不适合对三张或三张以上的图进行匹配。一个简单的解决方案是给定多张图，对这些图两两之间分别应用上面的算

法。但是这样会造成不一致的情况[62]。因此，这些方法不能很好地对多张主题图进行匹配。

为了解决这个问题，研究者们提出了一些针对多张图进行匹配的工作。Williams 等人[63] 对多张图的匹配问题进行了原理上的探讨。他们利用贝叶斯框架建立了一个推理矩阵（Inference Matrix），然后利用这个推理矩阵计算多图匹配的一致性。尽管他们给出的框架比较合理，但是他们并没有给出求解方案，因此这个方法难以应用在实际问题中。Ribalta 等人[64,65] 提出了计算多张图公共标签（Common Labeling）的算法，可以为多张图生成一张有代表性的公共图。这个算法通过一致的多图同构（Consistent Multiple Isomorphism）计算多张图的公共标签，可以处理三张及以上的图情况。但是，这个算法假设所有图的节点个数相同，使得算法的应用范围有较大局限性。2013 年，Yan 等人[62] 提出了一种基于有约束的整数二次规划（Integer Quadratic Programming, IQP）的多图匹配算法。给定 n 张图 G_1、G_2、……、G_n，该算法利用图两两之间的匹配算法计算相邻的图之间的匹配关系（即 G_k 与 G_{k+1}，$k \in [1, n-1]$），然后利用相邻图之间的匹配关系推算所有图之间的匹配关系，从而保证图匹配的一致性。为了确保生成的一致性匹配结果较优，Yan 等人将问题建模成带约束的整数二次规划求解。这个算法有两个局限性。首先，对于非相邻的两张图，该算法可能会漏掉一些图匹配结果。然后，IQP 的求解效率较低，因此算法难以用于需要系统实时响应的交互过程。

可视分析方法。 近年来，有一些工作利用图匹配技术对多张图进行可视比较。例如，Sambasivan 等人[66] 利用两张图之间的匹配算法对请求流（Request-flow）进行比较。他们用启发式算法提取出两张图之间的近似匹配结果。Hascoët 等人[67] 开发了一个结合了点线图与图匹配算法的交互式图匹配工具。该工具利用点的布局结果近似对点进行匹配。这个图匹配算法简单易实现。但是，因为点的布局并不是进行图匹配的可靠标准，这种算法可能会引入错误或者不确定性。另外，这个算法在布局效率与效果上也有不足之处。因为算法直接用力导向的布局方法，因此无法对太大的图进行快速布局。从效果上说，这个算法不能很好地从视觉上区分开来共有主题与独有主题。

通过上面的分析，我们可以看出，目前基于图匹配的方法在对多张图进行匹配，提取共有主题和独有主题时，在主题信息准确性方面有欠缺。另外，数据挖掘的方法没有可视化技术支持，可视化的方法布局效率不够高，因此在分析大量主题方面也有所欠缺。

2.3　多源动态文本主题分析的研究现状

现在多源动态文本分析的相关工作可以按照是否用了可视化技术分为文本挖掘方法以及可视分析方法两类。这两类方法的现有工作在提取主题及主题关键信息（领先-滞后关系）时，或者只考虑文本内容，或者只考虑词的时间序列，没有充分利用文本间引用关系等其他信息，因此主题以及领先-滞后关系的准确性还可以进一步提高。另外这些方法或者只考虑多个文本源之间同一个主题上的领先-滞后关系，或者只考虑两个文本源相关主题上的领先-滞后关系，无法有效分析多个文本源间相关主题的领先-滞后关系。因此，它们在主题信息丰富性方面也有不足之处。因为文本挖掘的方法没有可视界面，用户往往难以分析大量的主题。可视分析方法通过建立主题树与设计可视化交互界面，支持对大量主题的分析，可扩展性不错，但是在有效性上仍然有不足。表 2.3总结了这两类方法的优缺点。下面，我们对这两类方法的工作进行具体介绍与探讨。

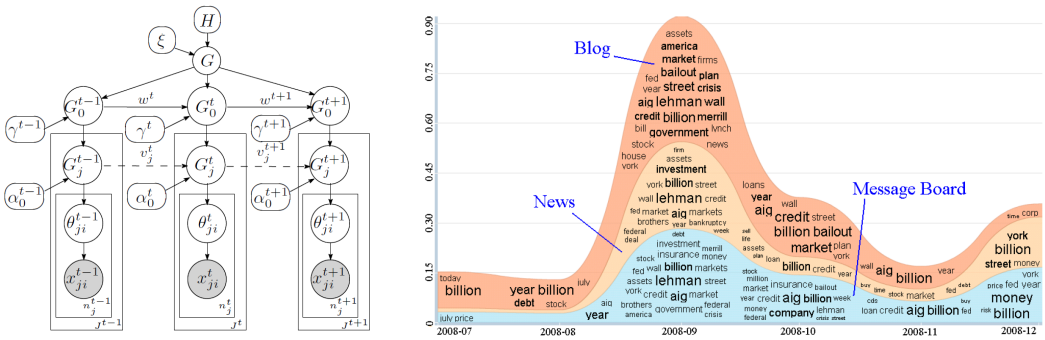
表 2.3　多源动态文本主题分析的研究现状。

	有效性		可拓展性
	主题信息准确性	主题信息丰富性	分析大量主题
文本挖掘方法	一般	一般	较差
可视分析方法	一般	一般	较好

2.3.1　文本挖掘方法

多源动态文本挖掘方法可以按照挖掘的主题信息分为三类。第一类方法主要挖掘不同文本源动态的共有与独有主题。第二、三类方法分别挖掘不同文本源的全局与局部领先-滞后关系。

动态共有与独有主题挖掘方法。 一个早期的多源动态文本挖掘的工作由 Wang 等人提出[68]。该方法认为不同文本源的主题模型几乎一样，唯一的区别是文本之间存在时间差。他们的研究重点在于如何在时间维度对不同文本源的文档进行对齐。当文档对齐以后，他们利用统一的主题模型对文档的主题进行建模。这个方法的关注重点在于不同文本源时间上的差异，对主题上的差异关注度不够，因此提取的准确性有不足之处。为了解决这个问题，Zhang 等人[69] 提出了 EvoHDP（Evolutionary Hierarchical Dirichlet Processes）方法。EvoHDP 的图模型与挖掘结果如图 2.7所示。该方法通过在 HDP 上加时间维度的依赖性，将提取多源静态文本共有主题的 HDP 方法拓展为了提取多源动态文本共有主题的 EvoHDP 方法。尽管 EvoHDP 在提取的主题比 Wang 等人的更准确，但是在提取独有主题方面还有不足之处。2011 年，Hong 等人[70] 提出了考虑时间依赖性的主题模型，用于挖掘不同文本源的动态共



(a) EvoHDP 图模型　　(b) EvoHDP 挖掘结果

图 2.7　EvoHDP 主题挖掘模型及结果。（图片引自 [69]）

有与独有主题。该方法忽略了不同文本源相关主题之间的领先-滞后关系，在主题信息丰富性上还可以进一步提高。

全局领先-滞后关系挖掘方法。 早期的领先-滞后关系挖掘方法主要研究主题的在所有时间点上全局的领先-滞后关系[71,72]。这些方法在三个不同的级别上对领先-滞后关系进行建模：短语级别、文档级别以及主题级别。短语级别的方法对每个弥因（Meme）[73]、专有名词[71] 或者词向量[74] 提取一个时间序列。然后，方法将来源于某文本源或者文本集合的时间序列在时间维度上前后平移，与来源于其他文本源的时间序列进行匹配。使得匹配程度最高（例如相关性最大[71,74] 或者时间序列峰值重叠[73]）的平移时间就认为是全局的领先时间。文档级别的方法通过概率模型检查一篇文档是否领先（或者影响）了其后出现的文档。其中用到的概率模型主要是生成式主题模型[75,76] 以及概率语言模型[77]。主题级别的方法[72,78] 将 LDA 主题模型融入到领先-滞后分析中，可以同时学习文档中的主题与主题之间的领先-滞后关系。尽管这些方法可以从一定程度上学习领先的或者最有影响力的文档集合与文档，它们无法学习领先-滞后关系随着时间的动态变化。

局部领先-滞后关系挖掘方法。 近来，研究人员提出了一些学习每个时间点局部领先-滞后关系的方法。TextPoineer[3] 通过计算不同时间点文档内容的相关性来计算局部的领先-滞后关系。具体来说，如果在 t 时刻，文档集合 A 的内容与文档集合 B 未来的内容更相似，而不是与 B 过去的内容更相似，那么认为文档集合 A 在 t 时刻是领先的。这个方法的局限性是只能检测同一个主题在不同文本集合中的领先-滞后关系，忽略了不同但是相关的主题之间的领先-滞后关系。Zhong 等人[79] 通过词的时间序列之间的协整（Cointegration）关系判断相关主题之间的领先-滞后关系。具体来说，对某文档集合中的一个词，他们都提取这个词出现频率的时间序列。如果两个时间序列的线性组合是一个稳态过程，则认为这两个词是有协整关系的（相关的）。这个方法的问题在于只能学习两个文本集合之间的领先-滞后关系。

现有数据挖掘的方法在提取主题的领先-滞后关系时考虑的文本信息较为单一，准确性不够令人满意。另外，目前还没有方法可以提取多个文本源的相关主题之间的领先-滞后关系。最后，因为这些方法大部分没有设计可视界面帮助用户分析大量主题，方法的可拓展性也有一些局限。

2.3.2　可视分析方法

多源动态文本可视分析方面的工作主要研究的是动态主题之间的关联。主题有多种可能随时间变化的关联。其中，最受关注的是主题的热度[80,81]、主题之间的竞争与合作关系[22,23] 以及主题的分裂、合并关系[1,21,82,83]。例如，EventRiver[81] 与 Leadline[80] 利用可视化的技术直观表现了文档类随时间的动态变化。近来，有一些研究人员重点研究了主题之间的合作[23] 与竞争[22] 关系。他们都利用随时间变化的河流来表现主题之间的关联。TextFlow[1] 用河流的方式展现主题动态的分裂、合并关系。RoseRiver 在 TextFlow 的基础上进行拓展，分析了层次化主题之间的分裂、合并关系。TopicStream[83] 在 RoseRiver 的基础上进一步拓展，分析了随时间不断到来的文本流中，主题之间的分裂、合并关系。ThemeDelta[21] 通过展示关键词如何合并到一个主题以及如何分裂到不同的主题，来分析主题发生转变的关键点。以上方法尽管可以分析一些主题之间的关联，但是在主题的领先-滞后关系方面做得还不够。

目前的可视分析方法中，只有 TextPioneer[3] 可以分析在同一个主题上，多个文本源之间的领先-滞后关系。该系统的问题是无法分析不同却相关的主题之间的领先-滞后关系。另外，它只用了基于词的文本内容对主题进行分析，忽略了文本的其他元信息（例如文档之间的引用关系等），因此主题内容以及主题的领先-滞后关系提取的准确性都可以进一步提高。

综上所述，现有可视分析的方法在可拓展性方面已经做得不错，但是在主题提取准确性以及主题信息丰富程度方面还不够令人满意。

第 3 章　单源动态文本的主题挖掘与可视分析

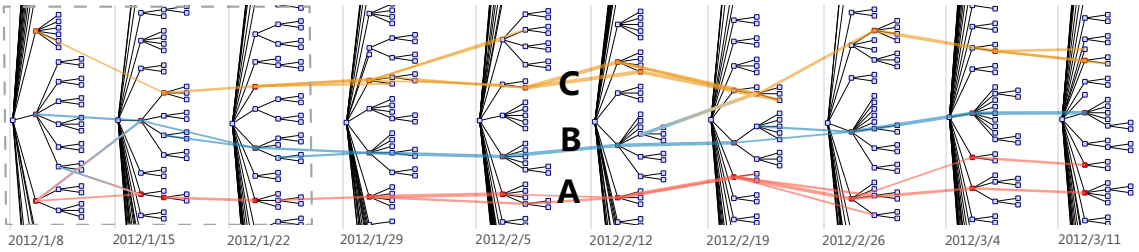
目前网络中存在着大量的单源动态文本，分析这些文本对人们做出合理决策有着重要的作用。例如，一个微软的公共关系分析师希望分析 2013 年上半年与微软相关的大事件——Xbox One 的发布、摩托罗拉对微软的起诉以及 Windows 8 的发布——如何随时间变化，以及产生了何种影响，从而帮助公司制定更好的公关策略。为了达到这个目的，这个公共关系分析师需要仔细分析动态新闻文本中相关的主题，并且理解这些主题的内容和影响如何随时间动态变化。

在很多应用中，主题被自然地组织成层次结构，而且这个层次结构随着时间动态变化[84]。目前，在分析随时间动态变化的层次结构方面有一些初步的工作。其中被广泛应用的是由 Chakrabarti 等人提出的动态凝聚式层次聚类[84]。该方法将不同时间点的主题组织成随时间动态变化的二叉树。但是，二叉树并不能非常准确地反映主题的层次结构，因为现实世界中主题的层次结构往往是多分枝的[2,5]。这种层次结构被称为多分枝树，树中每个中间节点可以有任意数目的孩子。有效地提取动态变化的多分枝主题树，对于帮助用户了解大量主题和主题之间的关系随时间的动态变化有着非常重要的作用。

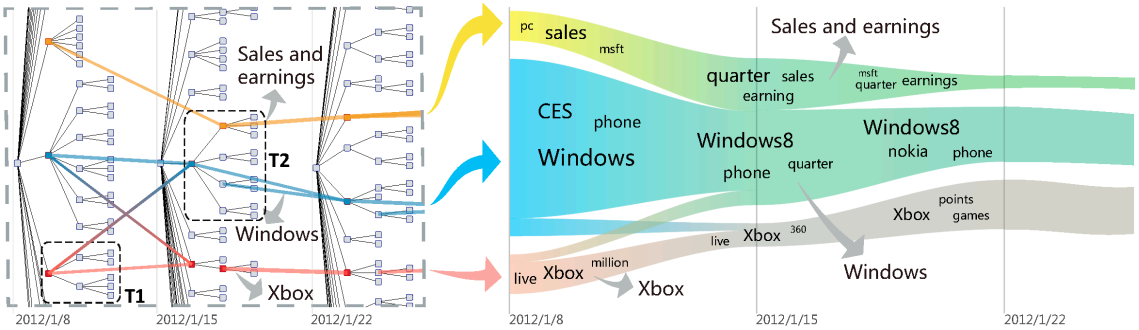
在这个章节中，我们定义并且研究了从动态文本中提取随时间变化的动态多分枝主题树的问题，并利用现有可视化技术辅助分析该动态多分枝主题树。这里，我们用一个新闻数据集来进行阐述。这个数据集中有 66,528 篇新闻。这些新闻是以“Microsoft”为关键词从必应新闻[85] 中检索出来的。图 3.1(a)显示了从这个数据集中提取的前十棵主题以及它们的层次结构。其中，高亮的三个主题分别是 Xbox 相关的主题（A）、Windows 相关的主题（B）以及销售与盈利相关的主题（C）。我们利用主题内容相似度提取出了不同树上相同主题的对应关系。通过显示对应关系的红边，我们可以看出在这期间，这三个主题相对稳定，只有较少的分裂、合并现象发生。通过提取这样动态变化的多分枝主题树并利用如图 3.1(a)所示的可视化形式展现，用户可以方便地

- 分析多分枝主题树随时间动态变化的规律以及它们内容的对应关系；
- 在树的任意层级分析用户感兴趣的主题以及它们随时间的分裂、合并关系。

为了更好地帮助用户理解他们感兴趣的主题以及这些主题随时间变化的规律，我们利用了一种动态主题可视化技术——TextFlow[1]——来显示主题之间动态的分裂、合并关系。如图 3.1(b)所示，TextFlow 用河流作为视觉隐喻来展现主题随时间的变化。其中，每条色带代表一个主题，横坐标代表时间。色带的宽度随横坐标不



(a) 2012 年 1 月 8 日至 3 月 17 日期间十棵动态变化的多分枝主题树。高亮的主题分别为 Xbox 相关的主题（A）、Windows 相关的主题（B）以及销售与盈利相关的主题（C）。



(b) 利用 TextFlow[1] 对 1 月 8 日至 1 月 28 日期间三个主题的分裂、合并关系进行可视化。

图 3.1　从与微软相关的必应新闻（2012 年 1 月 8 日至 7 月 21 日）中提取的多分枝主题树。我们将新闻文本以周为单位分为了 28 组。相应的，我们生成了 28 棵多分枝树。这些多分枝树的平均深度为 4，平均中间节点个数为 99，第一层的平均节点个数为 21。这里我们展示的是前十棵树。

断变化，展现出主题中包含的文档个数随时间的动态变化。就像现实生活中的河流一样，当一个主题渐渐分裂成多个主题时，一条主题流也分裂成多条支流；当多个主题渐渐合并时，相应的主题流也合并为一条。图 3.1(b)展现了 2012 年 1 月 8 日至 28 日期间，Xbox 相关主题、Windows 相关主题与销售及盈利相关的主题的分裂和合并关系。可以看到，与 Windows 相关的主题和与销售及盈利相关的主题在 1 月 15 日那一周合并在了一起。为了了解为什么主题会发生这样的合并，我们进一步研究了相应的新闻。我们发现，在那一周微软公布了自己的季度盈利报告。在对报告进行报导的新闻中，有很多新闻同时提到了 Windows 和微软的盈利状况。例如，有一篇新闻的标题是“Windows sales slowdown as Microsoft reports Q2 revenue up 5%”。随着对微软季度盈利报告报导的减少，这两个主题的相关性变低了，因此在下一周又分裂成了两个主题。另一个有趣的现象是第一周时 Windows 相关的主题中有一部分从该主题中分裂出去，在第二周与 Xbox 相关的主题进行了合并。导致这个现象的主要原因是 Windows NT 之父 Dave Culter 将他的注意力转向了 Xbox，希望能够使 Xbox 超越目前游戏平台的角色（“extend xbox beyond its status as a gaming platform”）。

受到上述例子的启发，我们希望生成一系列多分枝主题树。这些主题树既需

要有较好的平滑度，又需要有较高的拟合度。具体来说，如果当前文本数据没有与历史数据发生很大偏差，我们希望当前的主题树与上一棵主题树的内容和结构较为相似（平滑度）。另外，每棵主题树的结构要能够较好地反映相应时间点文档中的主题分布（拟合度）。

要生成满足上述两个条件的主题树非常具有挑战性。首先，生成动态变化的多分枝主题树并且对它们的变化规律进行建模并不容易。尽管目前最先进的建立多分枝树的方法[5,86] 可以提取有较高拟合度的多分枝主题树，但是这些方法无法保证树之间的平滑度。为了保证平滑度，最直接的想法是利用树上距离约束[84]。该约束将树上距离定义为树上两个节点之间通过树上的边连接而成的最短路径的长度。这样，树上距离约束可以通过最小化两棵树的树上距离的变化来提高树的平滑度。这种方法可以从一定程度上提高树的平滑度。但是它丢失了树的结构中重要的父亲-孩子关系，因此提取的树并不是最优的。另外，目前网上的文档（例如新闻）数目不断增长，文档量越来越大，在这种情况下保证算法效率并不容易。这是因为基于树的算法往往时间复杂度较高[86]，要生成一系列同时保证平滑度和拟合度的主题树往往耗时较长。

为了解决上面的问题，我们提出了一个建立随时间动态变化的贝叶斯多分枝树的方法：EvoBRT（Evolutionary Bayesian Rose Trees）。我们提出的方法既可以保证多分枝主题树的平滑度，又可以保证主题树的拟合度。具体来说，我们的主要贡献有如下三点。

首先，为了同时保证主题树的平滑度与拟合度，我们利用贝叶斯在线滤波框架（Bayesian Online Filtering）对动态多分枝树进行建模。在这个框架中，我们采用贝叶斯多分枝树算法（BRT）来建立多分枝主题树，从而保证生成的主题树拟合度高，方便用户对文档集合进行理解。我们的框架通过在 BRT 的基础上加入一个树结构的先验项来保证树的平滑度。这个树的先验项通过马尔科夫随机场（Markov Random Field, MRF）进行建模。这里的关键在于如何利用马尔科夫随机场的能量函数来衡量树的平滑度。现有研究成果表明，一棵树可以唯一地用一些三元组约束和扇形约束表示。如图 2.4(a)与图 2.4(b)所示，一个三元组约束是一棵含有三个叶子节点与两个中间节点的子树，一个扇形约束是一棵含有三个叶子节点与一个中间节点的子树。要生成一系列平滑变化的主题树，我们需要保证新生成的树上尽量保留了原来树上的三元组约束与扇形约束。因此，我们定义的树的不平滑程度与相邻的主题之间不相同的三元组约束与扇形约束个数正相关。

然后，我们进行了一系列实验，证明我们的方法在有效性和算法效率上都优于现有最先进的方法。具体来说，我们实现了两个动态多分枝树的算法，并且在多

种衡量标准下将它们和现有方法进行比较。这两个动态多分枝树的算法分别利用前一棵与前若干棵树作为约束。我们实现了三种基准算法与我们的方法进行比较。这三种基准算法中既包含建立多分枝树的算法，也包含建立二分枝树的算法。

最后，我们利用两个案例分析说明了我们方法的有效性和有用性。这两个案例分析着重说明了我们方法帮助用户在多个层次上分析文档集合的能力。

3.1　背景介绍：贝叶斯多分枝树

我们提出的 EvoBRT 在保证多分枝的拟合度方面主要是利用了目前最先进的建立多分枝主题树的算法：贝叶斯多分枝树。因此，我们在这里首先对贝叶斯多分枝树算法进行简要介绍。

3.1.1　算法流程简介

贝叶斯多分枝树的算法流程如表 3.1所示。在迭代开始之前，每个文档 $\mathbf{x}_i$ 被分到一个单独的子树 T_i 里：$T_i = \mathbf{x}_i$。这里，$\mathbf{x}_i$ 代表第 i 个文档的词特征向量。每个特征向量 $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(|\mathcal{V}|)}) \in \mathcal{R}^{|\mathcal{V}|}$，其中 $\mathcal{R}$ 代表实数集，$|\mathcal{V}|$ 代表文档中不同词的个数。在迭代中的每一步，算法将选取两棵子树 T_i 与 T_j，并将它们合并成子树 T_m。与二分枝树建立过程不同的是，贝叶斯多分枝树的建立过程中有三种不同的合并操作（图 3.2）：

- **Join**：$T_m = \{T_i, T_j\}$，这种情况下 T_m 有两个孩子节点。
- **Absorb**：$T_m = \{\text{children}(T_i) \cup T_j\}$，其中 children$(T_i)$ 指 T_i 的孩子节点集合，这种情况下 T_m 有 $|T_i| + 1$ 个孩子节点。

表 3.1　贝叶斯多分枝树算法。

算法一 贝叶斯多分枝树（BRT）

输入： 文本集合 $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$
对任意整数 $i \in [1, n]$，令子树 T_i 等于 $\{\mathbf{x}_i\}$
令 c 等于 n
while $c > 1$ **do**
1. 选择最大化下面表达式的 T_i 和 T_j 并把它们合并成 T_m

$$L(T_m) = \frac{p(\mathcal{D}_m|T_m)}{p(\mathcal{D}_i|T_i)p(\mathcal{D}_j|T_j)},$$

其中合并操作可能是如下三种中的一种：join, absorb, collapse。
2. 将 T_i 和 T_j 替换为 T_m
3. 令 c 等于 $c - 1$

end while

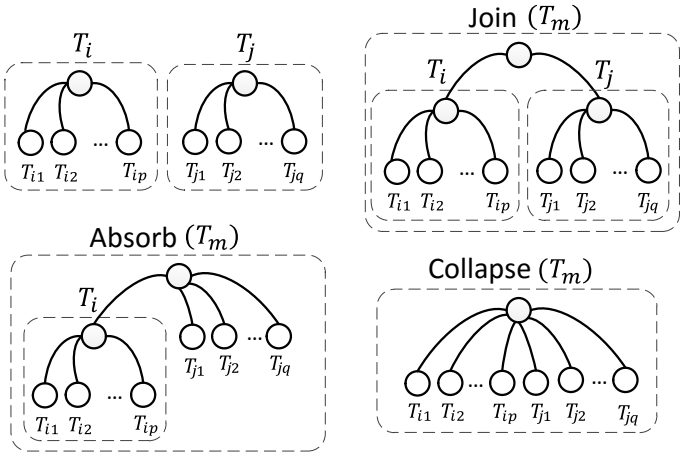


图 3.2　贝叶斯多分枝树建立过程中三种对子树进行合并的操作。

- **Collapse**：$T_m = \{\text{children}(T_i) \cup \text{children}(T_j)\}$，这种情况下 T_m 有 $|T_i| + |T_j|$ 个孩子节点。

具体来说，在迭代的每一步中，贝叶斯多分枝树算法贪心地找到两棵子树 T_i 和 T_j 以及一个合并操作，来最大化下面的似然概率增益值：

$$\frac{p(\mathcal{D}_m|T_m)}{p(\mathcal{D}_i|T_i)p(\mathcal{D}_j|T_j)} \tag{3-1}$$

这里 $p(\mathcal{D}_m|T_m)$ 是给定主题树 T_m，得到文档集 $\mathcal{D}_m$ 的似然概率。其中 $\mathcal{D}_m$ 是子树 T_m 所有叶子节点构成的集合，我们有 $\mathcal{D}_m = \mathcal{D}_i \cup \mathcal{D}_j$。概率 $p(\mathcal{D}_m|T_m)$ 是通过如下的表达式递归定义的：

$$p(\mathcal{D}_m|T_m) = \pi_{T_m} f(\mathcal{D}_m) + (1 - \pi_{T_m}) \prod_{T_i \in \text{children}(T_m)} p(\mathcal{D}_i|T_i) \tag{3-2}$$

这里 $f(\mathcal{D}_m)$ 是文档子集 $\mathcal{D}_m$ 的边缘概率（Marginal Probability）。π_{T_m} 是模型的混合比例（Mixing Proportion）。直观来说，π_{T_m} 代表 T_m 中所有文档在一个类，而不是被分为多棵子树的先验概率。在贝叶斯多分枝树中，这个先验概率定义如下：

$$\pi_{T_m} = 1 - (1 - \gamma)^{n_{T_m} - 1} \tag{3-3}$$

这里 $n_{T_m} = |\text{children}(T_m)|$ 表示 T_m 的孩子节点个数。$0 \le \gamma \le 1$ 是控制模型的超参数。参数 γ 越大，主题树上的类别划分得越粗糙；参数 γ 越小，主题树上的类别划分得越精细。

我们利用 DCM 分布[86,87] 计算文档的边缘概率 $f(\mathcal{D})$。具体公式如下：

$$f_{DCM}(\mathcal{D}) = \prod_{i}^{n} \frac{m_i!}{\prod_{j}^{|\mathcal{V}|} x_i^{(j)}!} \cdot \frac{\Delta(\boldsymbol{\alpha} + \sum_i \mathbf{x}_i)}{\Delta(\boldsymbol{\alpha})} \tag{3-4}$$

这里 $m_i = \sum_{j}^{|\mathcal{V}|} x_i^{(j)}$。$\boldsymbol{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(|\mathcal{V}|)})^T \in \mathcal{R}^{|\mathcal{V}|}$ 是控制狄利克雷分布的参数。这里的狄利克雷分布是每个类的多项分布的先验。

3.1.2　时间复杂度分析

上述自底向上算法的主要时间代价来源于以下两步：

- 每次迭代过程中寻找要合并的两棵子树；
- 计算合并两棵子树得到的似然概率增益（公式 (3-1)）。

假设目前有 c 棵子树，上述的两步的时间复杂度为 $O(c^2)$。在算法的开始，我们有 $c = n$ 棵子树。对于所有可能进行合并的子树对，我们首先利用公式 (3-1) 计算似然概率，然后在挑选下一步合并的子树对之前对这些子树对的似然概率增益进行排序。这里的时间复杂度为 $O(n^2 C_V + n^2 \log n)$。这里 C_V 是 $\mathbf{x}_i$ 中平均非零项的个数。高维文本数据中的 C_V 一般为几百，因此在分析时间复杂度时，它跟 $\log n$ 相比也不是一个可以忽略的量。另外，算法的空间复杂度是 $O(n^2)$。如果我们用 8 个字节的双精度浮点数来储存似然概率增益，考虑到 3 种不同的合并操作（Join, Absorb 与 Collapse），对于有 100,000 篇文档的数据集，算法需要占用 $3 \times 8 \times 10^{10} = 240G$ 字节的空间来存储所有子树对的似然概率增益。

3.2　动态多分枝主题树的建模

这一节中，我们首先介绍建立动态多分枝主题树的主要流程。然后，我们介绍确保平滑度的约束模型并说明如何将该模型拓展到多棵约束树的情况。最后，我们将对算法的时间复杂度进行分析。

3.2.1　贝叶斯在线滤波框架

我们的模型中假设文本数据顺序地不断到来。在第 t 个时间点，我们得到的文档集合记为 $\mathcal{D}^t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, ..., \mathbf{x}_{n_t}^t\}$，这里 n_t 是 t 时刻的文档篇数。我们假设存在一个潜在的主题树 T^t 能够较好地组织 t 时刻的文档。为了保证 T^t 能够体现文档的分布与文档的变化，我们将建立 T^t 的过程建模成一个贝叶斯在线滤波模型：

$$p(T^t|\mathcal{D}^t, T^{t-1}) \propto p(\mathcal{D}^t|T^t) p(T^t|T^{t-1}) \tag{3-5}$$

通过这个模型，t 时刻的树 T^t 的后验概率既依赖于它对当前数据的拟合程度 $p(\mathcal{D}^t|T^t)$，又依赖于条件先验概率 $p(T^t|T^{t-1})$。相应的，我们的模型考虑了动态层次聚类算法[84] 中最关键的两个量：拟合度与历史平滑代价（Historical Smoothness Cost）。这个模型只考虑前一棵树作为平滑约束。为了简单起见，我们先介绍一棵约束树的情况，然后介绍如何将模型拓展到多棵约束树。

因为有超指数（Super-exponential）个可能的 T^t 结构，直接最大化公式 (3-5) 中的 $p(\mathcal{D}^t|T^t)p(T^t|T^{t-1})$ 是不现实的。因此，我们遵循 BRT[5] 中的贪心凝聚聚类的方式来挑选每次迭代过程中需要合并的子树以及相应的合并操作（Join，Absorb 以及 Collapse）。在每次挑选过程中，我们选择能够最大化下面的后验概率增益的子树和合并操作：

$$\frac{p(\mathcal{D}_m^t|T_m^t)p(T_m^t|T^{t-1})}{p(\mathcal{D}_i^t|T_i^t)p(T_i^t|T^{t-1})\cdot p(\mathcal{D}_j^t|T_j^t)p(T_j^t|T^{t-1})} \tag{3-6}$$

这里 T_i^t 和 T_j^t 是进行合并的候选子树，T_m^t 是它们进行合并以后产生的子树，不同的合并操作产生 T_m^t 不同。$\mathcal{D}_i^t$ 和 $\mathcal{D}_j^t$ 分别代表 T_i^t 和 T_j^t 中包含的文档（叶子节点）的集合，我们有 $\mathcal{D}_m^t = \mathcal{D}_i^t \bigcup \mathcal{D}_j^t$。

高效计算公式 (3-6) 要求我们能够根据子树的拟合度与平滑度递归计算 T_m^t 的拟合度（$p(\mathcal{D}_m^t|T_m^t)$）与平滑度 $p(T_m^t|T^{t-1})$。贝叶斯多分枝树算法给出了拟合度的递归公式 (3-2)。我们参考非层次聚类上定义的马尔科夫随机场[88] 来定义平滑度递归公式。具体来说，我们将条件先验 $p(T^t|T^{t-1})$ 看作递归定义的马尔科夫随机场上的吉布斯（Gibbs）分布。这个马尔科夫随机场的能量函数可以认为是一系列平滑度损失的叠加。为此，我们首先需要衡量子树 T_i^t 和 T_j^t 合并成 T_m^t 以后的平滑度损失：

$$V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t) \tag{3-7}$$

利用这个平滑度损失，我们可以将子树的平滑度能量函数递归定义如下：

$$p(T_m^t|T^{t-1}) = p(T_m^t|T_i^t, T_j^t, T^{t-1})p(T_i^t|T^{t-1})p(T_j^t|T^{t-1}) \tag{3-8}$$

这里，

$$p(T_m^t|T_i^t, T_j^t, T^{t-1}) \triangleq \frac{1}{Z}\exp\Big(-\lambda V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t)\Big) \tag{3-9}$$

λ 是平衡拟合度与平滑度的参数，我们称之为约束项权重（Constraint Weight）。

$p(T_m^t|T^{t-1})$ 越高平滑度越好，$p(T_m^t|T^{t-1})$ 越低平滑度越差。利用公式 (3-8) 与公式 (3-9)，我们可以将公式 (3-6) 重写为

$$\frac{p(\mathcal{D}_m^t|T_m^t)}{p(\mathcal{D}_i^t|T_i^t)p(\mathcal{D}_j^t|T_j^t)} \cdot \frac{1}{Z}\exp\Big(-\lambda V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t)\Big) \tag{3-10}$$

上式中第一项可以利用贝叶斯多分枝树算法中的公式 (3-1) 进行计算。总的来说，算法流程图如表 3.2所示。下面我们主要探讨如何计算上式第二项中的平滑度损失 $V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t)$。

表 3.2　动态多分枝主题树构建方法。

算法一 动态多分枝主题（EvoBRT）树构建方法

输入： 文本集合 $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$
对任意整数 $i \in [1, n]$，令子树 T_i 等于 $\{\mathbf{x}_i\}$
令 c 等于 n
while $c > 1$ **do**
1. 选择最大化下面表达式的 T_i 和 T_j 并把它们合并成 T_m
$$\frac{p(\mathcal{D}_m^t|T_m^t)}{p(\mathcal{D}_i^t|T_i^t)p(\mathcal{D}_j^t|T_j^t)} \cdot \frac{1}{Z}\exp\Big(-\lambda V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t)\Big),$$
其中合并操作可能是如下三种中的一种：join, absorb, collapse。
2. 将 T_i 和 T_j 替换为 T_m
3. 令 c 等于 $c - 1$
end while

3.2.2　保证平滑度的约束模型

约束模型的主要目的是合理地衡量平滑度损失。合理地衡量平滑度对生成平滑变化的动态多分枝主题树非常关键。如果衡量方法对平滑度的损失估计有错漏，那么生成的多分枝树很可能只是在这种衡量方法下变化平滑，但是实际却并不平滑。例如，一种最直接的平滑度损失衡量方法是基于树上距离的衡量方法。这种衡量方法计算的是每两个叶子节点之间树上距离的平均变化。尽管这种方法能从一定程度上确保树的平滑，但是它没有考虑树的父亲-孩子关系。假设 T^{t-1} 和 T^t 如图 3.3所示，任何两个叶子节点之间在 T^{t-1} 上的树上距离都等于它们在 T^t 上的树上距离，但是 T^{t-1} 和 T^t 的结构却完全不同。因此，基于树上距离的衡量方法不能保证树的平滑度。

为了更合理地对约束模型进行建模，我们引入了三元组约束和扇形约束的概念。通过这两个约束衡量平滑度的损失，可以保证计算树层次结构的变化时没有

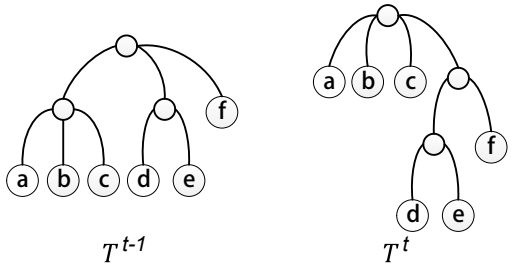


图 3.3　基于树上距离的平滑度损失衡量方法的不足之处：T^t 与 T^{t-1} 的层次结构差异很大，但是在这种衡量方法下计算的平滑度损失为 0。

错漏。为了快速计算打破三元组约束和扇形约束的个数，我们引入了约束树的概念，并介绍如何建立约束树、如何利用约束树快速计算打破三元组约束和扇形约束的个数。接下来，我们对算法的时间复杂度进行了分析。最后，我们探讨了如何把一棵约束树拓展到多棵约束树，从而考虑前若干个时间点的主题树层次结构。

3.2.2.1　三元组约束，扇形约束以及约束树

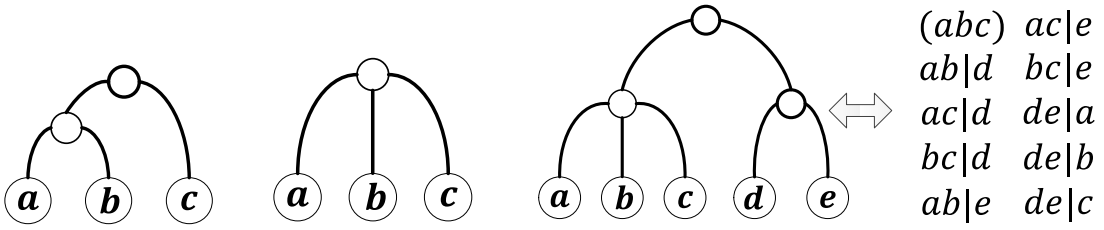
作为后续讨论的准备工作，我们首先介绍一下必要概念。

定义 3.1： 如图 3.4(a)所示，一个**三元组约束**是一个有三个叶子节点、两个中间节点的子树。假设三个叶子节点为 $\boldsymbol{a}$，$\boldsymbol{b}$ 和 $\boldsymbol{c}$，且 $ancestor(\boldsymbol{a},\boldsymbol{c})$ 是 $ancestor(\boldsymbol{a},\boldsymbol{b})$ 的祖先，我们将这个三元组约束记为 $\boldsymbol{ab|c}$。这里 $ancestor(\boldsymbol{a},\boldsymbol{b})$ 代表 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 最底层的共同祖先。

定义 3.2： 如图 3.4(b)所示，一个**扇形约束**是一个有三个叶子节点、一个中间节点的子树。假设三个叶子节点分别是 $\boldsymbol{a}$、$\boldsymbol{b}$ 和 $\boldsymbol{c}$，我们将这个扇形约束记为 $\boldsymbol{(abc)}$。

二分枝树中只有三元组约束，多分枝树上除了三元组约束，还有扇形约束。图 3.4(c)展示了一棵树和它含有的所有的三元组约束和扇形约束。这棵树含有九个三元组约束和一个扇形约束。

如下面的引理所示，Ng 等人[89] 证明了三元组约束和扇形约束含有了多分枝树的所有层次信息。



(a) 三元组约束 $ab|c$　(b) 扇形约束 (abc)　(c) 将树的层次结构表示成三元组约束与扇形约束

图 3.4　层次化聚类的两种层次约束以及如何将树的层次结构表示为这两种约束。

引理 3.1： 一棵多分枝树 T 可以被它含有的三元组约束和扇形约束唯一地定义。

因为三元组约束和扇形约束含有多分枝树的所有层次信息，我们用后一棵树打破的前一棵树的三元组约束和扇形约束的个数来衡量平滑度损失。一棵含有 n 个叶子节点的多分枝树含有 C_n^3 个三元组约束和扇形约束。因此，直接计算打破的三元组约束和扇形约束的个数通常需要 $O(n^3)$ 的内存和 $\Omega(n^3)$ 的时间复杂度，内存和时间上的消耗都很大。为了解决这个问题，我们用一棵树来组织所有的三元组约束和扇形约束。我们称这棵树为约束树。

定义 3.3： **约束树** $\tilde{T}^t$ 组织了所有从文档集 $\mathcal{D}^t$ 中提取的三元组约束和扇形约束。它基于前一棵树 T^{t-1} 进行初始化，并且随着三元组约束和扇形约束被打破进行不断调整。

在下面的两个小节，我们将分别介绍约束树是如何初始化并且进行调整的。

3.2.2.2　约束树的初始化

建立约束树 $\tilde{T}^t$ 的基本思想是将文档集 $\mathcal{D}^t$ 中的文档对应到前一棵树 T^{t-1} 中最相关的主题上。对于不属于前一棵树 T^{t-1} 上任何主题的文档，我们将不加入 $\tilde{T}^t$ 中，直接在 T^t 生成新的主题。在我们的方法中，我们实现了两种计算 $\mathbf{x}_i^t$ 和 $\mathcal{D}_m^{t-1}$ 的相关性的方法。第一种方法是基于文档之间的余弦相似度的：

$$sim_{cos}(\mathbf{x}_i^t, \mathcal{D}_m^{t-1}) \triangleq \cos(\mathbf{x}_i^t, \sum_{\mathbf{x}_j^{t-1} \in \mathcal{D}_m^{t-1}} \mathbf{x}_j^{t-1}) \tag{3-11}$$

第二种计算相关性的方法是预测衡量法。这个方法计算的是给定 $\mathcal{D}_m^{t-1}$ 时，得到 $\mathbf{x}_i^t$ 的条件概率[90]:

$$\begin{aligned} sim_{pred}(\mathbf{x}_i^t, \mathcal{D}_m^{t-1}) &\triangleq \log p(\mathbf{x}_i^t | \mathcal{D}_m^{t-1}) \\ &= \log \int_\theta p(\mathbf{x}_i^t | \theta) p(\theta | \mathcal{D}_m^{t-1}) d\theta \end{aligned} \tag{3-12}$$

要将文档 $\mathbf{x}_i^t$ 对应到 T^{t-1} 上最相关的子树（主题）上，我们采用自顶向下的贪心搜索算法。具体来说，如果 $\mathbf{x}_i^t$ 跟某父亲节点的相关程度大于它跟父亲节点的所有孩子节点的相关程度，并且大于预先设定的相关值 s_0，我们将停止搜索过程，选择父亲节点作为最相关的主题；否则，我们将相关度最大的孩子节点作为下一轮的父亲节点，并且重复上面的过程。将 $\mathcal{D}^t$ 中所有的文档 $\mathbf{x}_i^t$ 对应到前一棵树 T^{t-1} 上以后，我们得到了新的约束树 $\tilde{T}^t$，也就是初始化的约束树。

3.2.2.3　约束树的修改操作及其代价

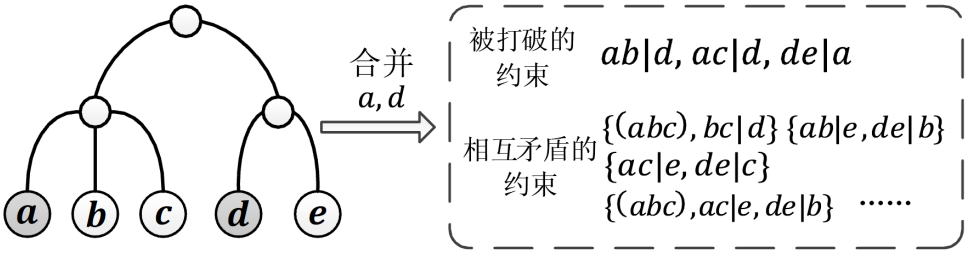


图 3.5　两种状态的约束：被打破的约束（Violated Constraints）和互相矛盾的约束（Conflicting Constraints）。

有了初始化的约束树以后，一种最直接的计算平滑度损失的方法是在建立 t 时刻的树 T^t 时，计算子树 T_i^t 和 T_j^t 合并以后打破了多少约束树上的三元组约束和扇形约束。尽管这个方法非常自然直接，它的一个主要问题是会产生互相矛盾的约束。这是因为我们在计算平滑度损失时忽略了一些三元组约束和扇形约束。例如，给定如图 3.5 所示的约束树 $\tilde{T}^t$，如果我们合并 a 和 d，将造成两种新的状态的约束。第一种是被打破的约束。在这个例子中，有三个三元组/扇形约束被打破了。以三元组约束 $ab|d$ 为例，这个三元组约束规定 a 和 b 需要先进行合并，才能和 d 进行合并。但是，因为 a 和 d 先合并了，这个三元组约束被打破了。第二种是相互矛盾的约束，即无法在约束树上共存的约束。以扇形约束 (abc) 为例，如果 a 和 d 进行了合并，(abc) 和 $bc|d$ 是互相矛盾的，因为它们不可能同时在 T^t 上被满足了。

除了被打破的约束以外，相互矛盾的约束也导致了平滑度的损失。但是，计算相互矛盾的约束打破的平滑度的损失并不容易，因为相互矛盾的约束之间的关系非常复杂。例如，一个约束可能和多个约束相互矛盾。另外，除了两两之间相互矛盾的约束，还有可能三个约束之间存在矛盾，甚至超过三个约束之间存在矛盾。如图 3.5所示，如果 a 和 d 进行了合并，尽管 (abc)、$ac|e$ 和 $de|b$ 这三个约束两两之间都不矛盾，但是这三个约束并不能同时在 T^t 上被满足了。

通过上面的分析，我们可以看到正确衡量矛盾约束产生的平滑度损失非常困难。为了解决这个问题，我们引入了约束树上的两种基本修改操作：**MERGE**（融合）以及 **SPLIT**（分离）。图 3.6中对这两种操作进行了简要的说明。下面，我们对这两种操作进行正式的定义。

定义 3.4： **MERGE** 是指子树 $\tilde{T}_k$ 将它的数据和孩子节点传递给它的父亲。然后，$\tilde{T}_k$ 将被从约束树上删除。

定义 3.5： **SPLIT** 是指将子树 $\tilde{T}_l'$ 的部分孩子节点独立出来，变成一棵新子树 $\tilde{T}_k$ 的孩子节点，并且将这棵新子树 $\tilde{T}_k$ 加入到原子树 $\tilde{T}_l'$ 的孩子中。

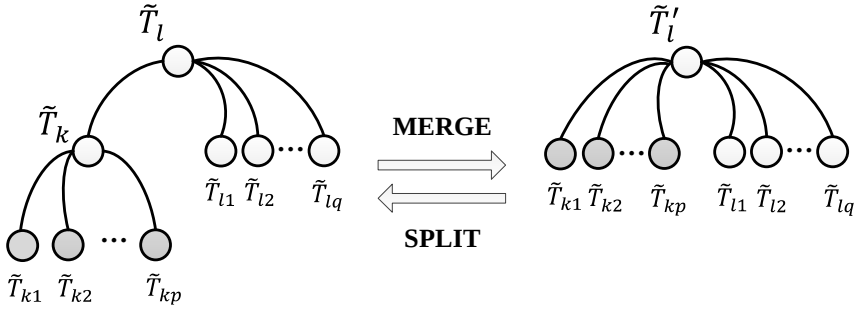


图 3.6　约束树上的两种修改操作：**MERGE**（融合）以及 **SPLIT**（分离）。

接着，我们说明为什么这两个操作可以消除被打破的约束和矛盾的约束，生成一致的约束树。简单起见，我们首先以两层的树为例来阐述基本思想。如图 3.2所示，贝叶斯多分枝树提供了三种对子树进行合并的操作：Join, Absorb 和 Collapse。如果子树 T_i 和 T_j 进行合并利用的操作与约束树上的操作不符，将会引入被打破的约束，或者造成互相矛盾的约束。为了避免这种情况，我们可以更新约束树，从而使得它跟目前数据的层次结构相符。例如，假设子树 T_i 和 T_j 是用 Absorb 操作（图 3.7A）进行合并的。如果约束树上是用 Join 操作进行合并的（图 3.7B），将会产生被打破的约束或者互相矛盾的约束。为了解决这个问题，我们利用一个 **MERGE** 操作将约束树转化成如图 3.7A 的形式。更多的利用 **MERGE** 和 **SPLIT** 修改约束树的方法见图 3.7。这个图说明我们可以通过 **MERGE** 和 **SPLIT** 操作将约束树进行任意形态地转化。因此，这两个操作足够我们维护一棵一致的约束树。

有了 **MERGE** 和 **SPLIT** 操作，我们可以通过计算这两个操作中打破的三元组约束和扇形约束来计算平滑度损失。下面的定理说明了相应的计算方法。

定理 3.1： 在 **MERGE** 操作中，只有三元组约束会被打破，而且被打破的三元组约

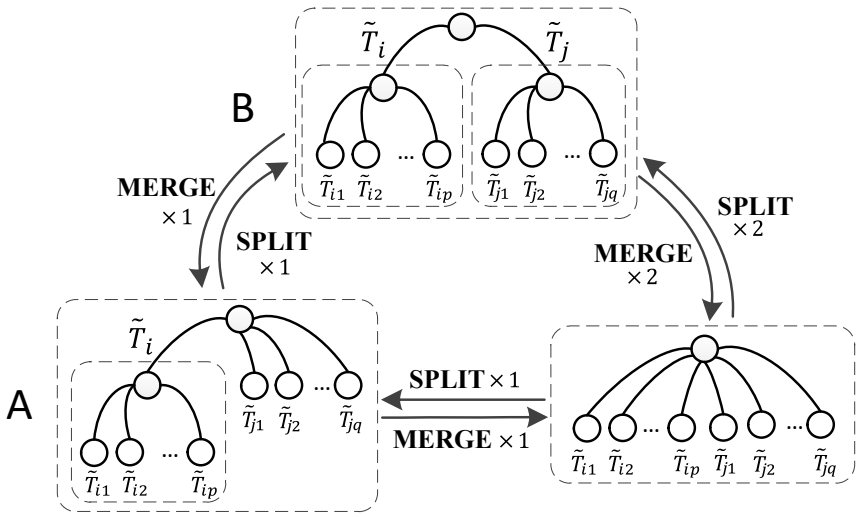


图 3.7　利用 **MERGE**（融合）以及 **SPLIT**（分离）更新约束树。

束都会转化成扇形约束。被打破的三元组约束的个数是

$$V_{\mathbf{MERGE}}(\{\tilde{T}_k, \tilde{T}_l\} \to \tilde{T}_l') = \frac{|\tilde{\mathcal{D}}_k|^2 - \sum |\tilde{\mathcal{D}}_{ki}|^2}{2}(|\tilde{\mathcal{D}}_l| - |\tilde{\mathcal{D}}_k|) \tag{3-13}$$

这里 $|\tilde{D}_k|$ 是子树 $\tilde{T}_k$ 中的叶子个数。

证明 给定一棵树 T，里面的扇形约束的个数为

$$|\mathcal{F}_T| = \sum_{\substack{T_S \in \text{sub-trees}(T) \\ |\text{children}(T_S)|>2}} \sum_{\substack{T_{si}, T_{sj}, T_{sl} \in \\ \text{children}(T_S)}} |\mathcal{D}_{si}||\mathcal{D}_{sj}||\mathcal{D}_{sl}| \tag{3-14}$$

这里，$\mathcal{F}_T$ 是 T 中扇形约束的集合，$|\mathcal{F}_T|$ 是集合中扇形约束的个数。

因为 $\tilde{T}_l'$ 中所有的三元组约束都包含在 $\tilde{T}_l$ 中，**MERGE** 操作仅仅打破了三元组约束，把它们都改变成了扇形约束。因此，**MERGE** 操作的代价就等于 $\tilde{T}_l$ 和 $\tilde{T}_l'$ 中扇形约束个数差。

$$|\mathcal{F}_{\tilde{T}_l'}| - |\mathcal{F}_{\tilde{T}_l}| = \frac{|\tilde{\mathcal{D}}_k|^2 - \sum |\tilde{\mathcal{D}}_{ki}|^2}{2}(|\tilde{\mathcal{D}}_l| - |\tilde{\mathcal{D}}_k|) \tag{3-15}$$

□

相似地，我们可以证明如下有关 **SPLIT** 操作的定理：

定理 3.2： 在 **SPLIT** 操作中，只有扇形约束会被打破，而且被打破的扇形约束都会变成三元组约束。被打破的扇形约束的个数是：

$$V_{\text{s}}(\tilde{T}_l' \to \{\tilde{T}_k, \tilde{T}_l\}) = \frac{|\tilde{\mathcal{D}}_k|^2 - \sum |\tilde{\mathcal{D}}_{ki}|^2}{2}(|\tilde{\mathcal{D}}_l| - |\tilde{\mathcal{D}}_k|) \tag{3-16}$$

上面假设被合并的子树已经是相邻的了。在实际应用中，要合并的两棵子树并不一定是相邻的。这种情况下，我们首先利用 **MERGE** 操作将这两棵子树移到它们最底层的共同祖先下。一旦它们被移到了同一个祖先下，我们再利用 **MERGE** 或者 **SPLIT** 操作将相应的约束树修改成一致的。最后，被打破的三元组约束和扇形约束个数可以通过将所有 **MERGE** 和 **SPLIT** 的代价相加进行计算：

$$V_{T^{t-1}}(\{T_i^t, T_j^t\} \to T_m^t) = \sum_l V_{\text{OPT}_l} \tag{3-17}$$

这里 OPT_l 是 **MERGE** 或者 **SPLIT** 操作。图 3.8展示了对约束树进行修改的过程。

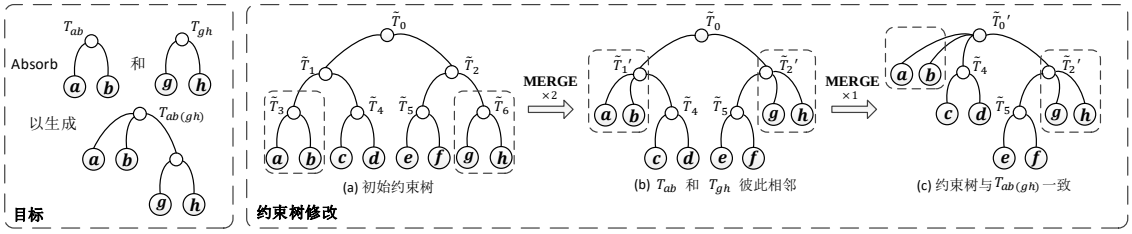


图 3.8　修改约束树的例子。

表 3.3　动态多分枝主题树算法的时间复杂度。其中，n 是当前时间点的文档个数，C_V 是文本特征向量中非零元素个数，K 是考虑的近邻个数，h 是多分枝主题树的高度，d 是 SpillTree 算法中特征向量降维后的维度。初始化约束树 $\tilde{T}^t$ 的代价是 $O(nC_V \log n)$。

	贝叶斯多分枝树的构建	计算平滑度损失
EvoBRT	$O(n^2 C_V + n^2 \log n)$	$O(n^2 h \log n)$
KNN-EvoBRT	$O(n^2 C_V + n^2 \log K)$	$O(nKh \log n)$
SpillTree-EvoBRT	$O(ndC_V + nd \log n \log K)$	

3.2.3　时间复杂度分析

EvoBRT 算法的时间复杂度可以分为三个部分：贝叶斯多分枝树的构建、约束树的初始化以及平滑度损失的计算。表 3.3总结了算法的时间复杂度。

Liu 等人提出了几种贝叶斯多分枝树的快速实现方法[86]。在本论文中，我们利用这些方法来建立多分枝主题树。我们用 EvoBRT 来表示基于原始贝叶斯多分枝树实现的方法，用 KNN-EvoBRT 和 SpillTree-EvoBRT 来表示基于 Liu 等人的快速近似算法实现的方法。为了简单起见，我们将 KNN-EvoBRT 和 SpillTree-EvoBRT 统称为近似动态算法。这三种算法构建贝叶斯多分枝树的时间复杂度结果如表 3.3所示。感兴趣的读者可以参考 Liu 等人的论文[86] 了解更多细节。

为了建立初始的约束树 $\tilde{T}^t$，我们将文档集 $\mathcal{D}^t$ 中每篇文档映射到 T^{t-1} 上的一个合适主题。因为我们需要自顶向下地寻找最合适的主题，这部分的时间复杂度是 $O(nC_V \log n)$。

计算平滑度损失的时间复杂度主要来源于两个部分。第一部分主要计算打破约束的个数，这部分的时间复杂度在 EvoBRT 算法中是 $O(n^2 h)$，在近似动态算法中是 $O(nKh)$。这里 h 是 $\tilde{T}^t$ 的高度，K 是考虑的近邻个数。当约束树更新以后，一些后验概率增益也被改变了。第二部分的时间复杂度来源于更新后验概率增益。这部分的时间复杂度在 EvoBRT 算法中是 $O(n^2 h \log n)$，在近似动态算法中是 $O(nKh \log n)$。这里 $\log n$ 的因子体现的是更新排好序的列表中的一个元素的代价。$n^2 h$ 和 nKh 指的是在 EvoBRT 的算法和近似动态算法中进行后验概率增益更新的最大子树对个数。

3.2.4　拓展：多棵约束树

在这一节中，我们将介绍如何将模型进行拓展，从而将条件先验概率修改成 $p(T^t|T^{t-1},T^{t-2},...)$，达到考虑多棵约束树的目的。要拓展成多棵约束树，其中最大的问题在于不同的约束树之间存在相互矛盾的约束。这种相互矛盾的约束可能会破坏树的平滑度。一般来说，越多的约束树可能引入相互矛盾的约束越多（见实验部分）。为了解决互相矛盾的约束的问题，我们通过基于图编辑距离的图匹配[54]方法把不同的约束树进行匹配，从而提取出一棵匹配后的没有互相矛盾约束的子树。具体来说，给定 i 棵约束树 $\{\tilde{T}^{t-i+1},...,\tilde{T}^{t-1},\tilde{T}^t\}$，我们首先对 $\tilde{T}^{t-i+1}$ 和 $\tilde{T}^{t-i+2}$ 进行匹配，匹配结果记为 $M(t-i+1,t-i+2)$。然后，我们将 $M(t-i+1,t-i+2)$ 与 $\tilde{T}^{t-i+3}$ 进行匹配，得到 $M(t-i+1,t-i+2,t-i+3)$。接着，我们重复上面的匹配过程，得到最终的结果 $M(t-i+1,...,t-1,t)$。最后，我们将这个最终的匹配结果当作初始的约束树输入到我们的动态聚类算法中，进行后续的计算。

3.3　数值实验

为了验证我们的算法，我们在几个真实数据集上进行了一系列的实验。接下来，我们首先介绍实现的几个基准算法。然后，我们在 20NewsGroup 数据集[91] 上验证了我们约束模型的有效性。接着，我们展示我们的算法生成的动态多分枝树既有较好的平滑度，又有较高的拟合度。接下来，我们说明了我们的动态多分枝算法和原来不考虑平滑度的非动态算法速度几乎一样快。最后，我们研究了多棵约束树如何影响树的平滑度。实验结果证明我们的方法与基准算法相比，在平滑度、拟合度以及算法效率上都有优势。

在我们的实验中，没有特别说明的情况下，我们都用 KNN-EvoBRT（$K=50$）这个算法。这是因为 KNN-EvoBRT 算法效率高，而且相比原始算法在有效性上也相当[86]。在每个实验中，我们都利用网格搜索（Grid Search）找到让贝叶斯多分枝树的似然概率最高的参数，用这个参数建立动态多分枝主题树，记录相应的结果。

3.3.1　基准算法

实验部分中，我们希望验证提出的动态多分枝主题树算法以及设计的约束模型的有效性和效率。为了达到这个目的，我们基于 Chakrabarti 等人提出的动态层次聚类算法[84] 和 Heller 等人的贝叶斯层次聚类算法[92] 实现了三个基准算法。表 3.4显示了这三种基准算法的建树方法以及约束模型。

我们用贝叶斯层次聚类方法建立二分枝的动态主题树是因为，相比传统的凝聚式聚类算法，它生成的层次结构更准确，结构也更平衡[92]。基于树上距离的约

表 3.4　基准算法。

	建树方法	约束模型
BinaryDistance	动态二分枝方法	基于树上距离的方法
BinaryOrder	动态二分枝方法	基于三元组约束和扇形约束
MultiDistance	动态多分枝方法	基于树上距离的方法

束模型是来自于 Chakrabarti 等人的提出的动态层次聚类方法[84]。我们将这个方法融入到 BinaryDistance 和 MultiDistance 这两个基准算法中。具体来说，基于树上距离的约束模型采用如下的公式衡量平滑度：

$$\log p_{Dist}(T^t|T^{t-1}) \triangleq -\lambda E_{\substack{r,s\in leaves(T^t)\\ r\neq s}}(d_{T^t}(r,s) - d_{\tilde{T}^t}(r,s))^2 \tag{3-18}$$

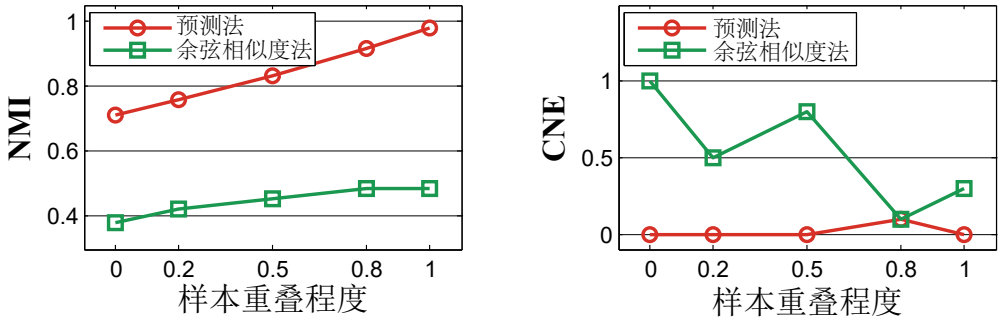
这里，$d_{T^t}(r,s)$ 指的是 r 和 s 在 T^t 上的树上距离，λ 是平衡平滑度和拟合度的约束项权重。我们用 Chakrabarti 等人提出的平方启示法来最大化平滑度，因为这种方法很容易拓展成多分枝树的情况。

3.3.2　约束模型有效性实验结果

约束模型有效性实验有两个主要目的。第一个目的是检验两种初始化约束树方法的有效性，并且分析哪个方法更好。第二个目的是验证我们的约束模型的聚类质量。

3.3.2.1　实验设置

这个实验中，我们用到的是 20NewsGroup[91] 的数据集。这个数据集中有棵人工标注的两层的主题树。这个主题树上，第一层有 7 个类，第二层有 20 个类。我们将只有一个孩子节点的父亲节点去掉，最后得到了一棵第一层为 4 个类，第二层为 17 个类的主题树。在每次试验过程中，我们随机采样出来 2,000 篇文档，并按照标注结果建立出来正确的多分枝主题树。我们将这棵树当作 $t-1$ 时刻的主题树 T^{t-1}。然后，我们重新采样 2,000 篇文档，将这些文档映射到 T^{t-1} 上，得到我们模型中初始化以后的约束树。我们可以通过合理采样，保证这部分文档和 T^{t-1} 上的文档有一定程度的重叠。在本实验中，我们考虑了从 0 到 1 的五种不同的重叠程度。对其中的每一种重叠程度，我们都进行了五次不同的采样，然后把五次采样的实验结果进行平均作为最终的实验结果。这里，公式 (3-3) 中贝叶斯多分枝树的值设为 0.1，$\alpha^{(i)} = 0.01$ $(i = 1, ..., |\mathcal{V}|)$。



(a) NMI 随重叠程度的变化曲线。　　(b) CNE 随重叠程度的变化曲线。

图 3.9　初始化约束树的在不同衡量标准下的聚类质量。

3.3.2.2　评价标准

我们利用两种方法来评价聚类的质量：归一化互信息 NMI（Normalized Mutual Information）和聚类个数误差 CNE（Cluster Number Error）。NMI[93] 是一种应用广泛的衡量聚类质量的评价指标。NMI 在大部分情况下能很好衡量聚类质量，但是当两个聚类结果的类别个数相差比较大时，NMI 尚有不足之处。我们用一个例子作为说明。假如人工标注的聚类结果中有 50 类，每类里面有 20 个样本，而算法聚类结果有 1000 类，每类只有一个样本，这时聚类结果并不理想，但是 NMI 的值却很高（0.75）。为了解决这个问题，我们引入了 CNE 这个评价标准，来衡量算法生成的聚类结果与人工标注结果类别个数的差异。CNE 的值越大，聚类准确性越低。因为我们这里考虑的是层次聚类，我们将计算每一层的聚类结果的 NMI 与 CNE 值，然后将它们进行平均，得到最终的结果。总的来说，NMI 值越大，CNE 值越小，聚类准确性越高。

3.3.2.3　结果

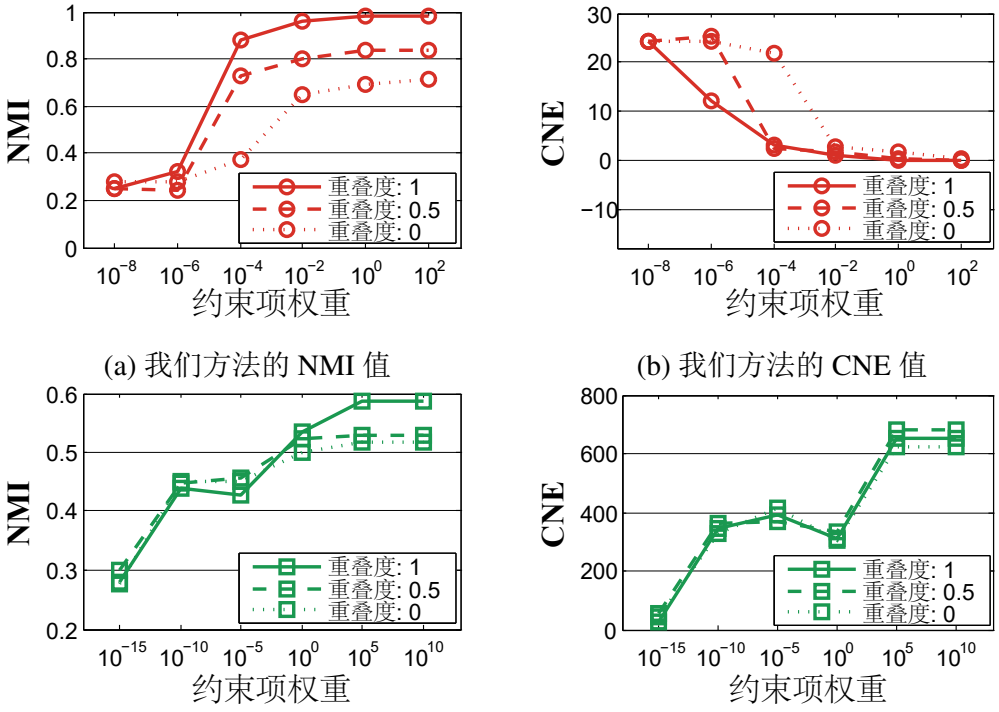
首先，我们测试了哪种计算初始化约束树的方法效果更好。我们将第 3.2.2.2节中用到余弦相似度的初始化方法称为余弦相似度法（公式 (3-11)），将预测文档在主题中出现概率的方法称为预测法（公式 (3-12)）。图 3.9显示在不同衡量标准下，初始化约束树的聚类质量如何随样本重叠程度变化。图中可以看出，预测法在两种衡量标准下皆优于余弦相似度法。即便在样本重叠程度为 0 的情况下，预测法生成的初始约束树的 NMI 也高达 0.7。预测法的结果更好可能是因为我们的建树方法是基于概率的，因此同样基于概率的预测法与我们的模型更加吻合。

然后，我们将提出的约束模型与基准算法的约束模型进行比较，从而说明我们的约束模型可以得到更高质量的聚类结果。这个实验中，我们只用到了 MultiDistance 这个聚类算法。另外两个基准算法 BinaryDistance 和 BinaryOrder 没在这里进

行测试，是因为它们只能建立二分枝树，而本实验中的人工标注层次结构是多分枝的。图 3.10显示了我们算法与 MultiDistance 聚类方法的聚类质量在不同的样本重叠程度时如何随约束项权重 λ 变化。图中可以看出，我们的算法远优于基准算法 MultiDistance。当样本重叠程度是 1 时，我们可以保证还原出人工标注的层次结构，但是 MultiDistance 却无法还原人工标注的层次结构（NMI 的值小于 0.6）。即使在样本重叠程度是 0 的情况下，我们的方法的 NMI 值仍然较高（为 0.7），CNE 值仍然很小（接近 0）。然而，MultiDistance 的方法在样本重叠度为 0 的情况下 NMI 值为 0.5，CNE 非常大（为 650），远远差于我们的方法。当用我们的算法时，随着约束项权重增加，NMI 和 CNE 的值都越来越好。但是对于 MultiDistance，尽管 NMI 值随约束项权重增加变好，但是 CNE 的值却越来越差。这体现出基于树上距离的约束模型（即 MultiDistance 的约束模型）有将树的层次结构变得越来越平的趋势。

3.3.3　平滑度与拟合度实验

本实验的目的是为了验证我们的算法可以同时保证动态多分枝树的平滑度和拟合度。



(a) 我们方法的 NMI 值　(b) 我们方法的 CNE 值

(c) MultiDistance 方法的 NMI 值　(d) MultiDistance 方法的 CNE 值

图 3.10　我们约束模型和基准算法 MultiDistance 的约束模型聚类质量对比。

3.3.3.1　实验设置

在这个实验中，我们用到的是从 2006 年 1 月到 2007 年 6 月的纽约时报的数据[94]。这个数据集含有 12,798 篇与艺术、时尚、旅游、商业和体育相关的新闻。我们将这个数据按照时间分为 9 组，每组中含有两个月的新闻。我们从每个数据集中随机采样出 1,000 篇新闻。为了尽量消除由采样带来的随机性，我们将数据随机采样 5 次，并将这 5 次实验的平均结果作为最终的实验结果。贝叶斯多分枝树的参数 γ 和 $\alpha^{(i)}$ 分别设为 0.03 和 0.0005。

3.3.3.2　评价标准

在这个实验中，我们用似然概率（Likelihood）来衡量树的拟合度。对于树的平滑度，我们引入了三种衡量方法。

- 基于三元组约束和扇形约束的平滑度 $\mathbf{S_{Order}}$。这个平滑度来自于我们算法。它的定义如下：

$$S_{Order} = \frac{1}{\lambda}\log(p(T^t|T^{t-1})) \tag{3-19}$$

  这里 $p(T^t|T^{t-1})$ 根据公式 (3-8) 递归计算。当前树与上一棵树中不一致的三元组约束和扇形约束个数越少，S_{Order} 值越大，平滑度越高。
- 基于树上距离的平滑度 $\mathbf{S_{Dist}}$。这个平滑度来自于基准算法。它通过计算任意叶子节点之间树上距离的变化衡量树结构的变化：

$$S_{Dist} = \frac{1}{\lambda}\log(p_{Dist}(T^t|T^{t-1})) \tag{3-20}$$

  这里，$p_{Dist}(T^t|T^{t-1})$ 根据公式 (3-18) 定义。
- Robinson-Foulds 平滑度 $\mathbf{S_{RF}}$。这个平滑度利用了被广泛应用于衡量种系树之间距离的方法——Robinson-Foulds 距离[95]：

$$S_{RF} = 1 - \frac{d_{RF}(T^t, T^{t-1}(\mathcal{D}^t)) + d_{RF}(T^{t-1}, T^t(\mathcal{D}^{t-1}))}{2} \tag{3-21}$$

  这里，$T^{t-1}(\mathcal{D}^t)$ 代表利用 t 时刻文档 $\mathcal{D}^t$ 与 $t-1$ 时刻多分枝树 T^{t-1} 建立的约束树。S_{RF} 即 $(T^t, T^{t-1}(\mathcal{D}^t))$ 与 $(T^{t-1}, T^t(\mathcal{D}^{t-1}))$ 之间距离的平均值。这里，我们实现的是 Robinson-Foulds 距离的一个改进版本[96]。

3.3.3.3　结果

利用我们的算法以及 MultiDistance，我们分别建立了两棵动态变化的多分枝树，它们的平均树深为 5。利用 BinaryDistance 和 BinaryOrder，我们分别建立了两棵动态变化的二分枝树，它们的平均树深为 366。图 3.11中对比了这四种方法生成的动态树的平滑度与拟合度。我们采用网格搜索的方法，对八种不同的约束项权重进行了遍历。因为不同的约束模型需要的参数不同，我们在这里用了两组不同的约束项权重。具体来说，我们的算法和 BinaryOrder 算法的约束项权重是 $\{3e^{-6}, 1e^{-5}, ..., 3e^{-3}, 1e^{-2}\}$。MultiDistance 和 BinaryDistance 的约束项权重是 $\{3e^{-6}, 1e^{-5}, ..., 3e^{-3}, 1e^{-2}\}$。约束项权重越大，平滑度所占的重要性越高。在每张图中，我们将相同算法在不同约束项权重时得到实验结果连成一条线。基于对图中结果的分析，我们得到了下面的结论。

首先，我们的算法生成的动态树平滑度远高于基准算法生成的动态树。与此同时，我们生成的动态树拟合度也更高。除了在 S_{Order} 这个平滑度指标上效果很好以外，我们在 S_{Dist} 与 S_{RF} 这两个我们约束模型没有直接优化的平滑度指标上结果也更好。这是因为三元组约束和扇形约束包含了多分枝树上层次结构的所有信息。因此，基于这个平滑度约束的方法可以同时优化平滑度与拟合度。MultiDistance 在 S_{Dist} 这个指标下结果还不错。但是，它在 S_{Order} 和 S_{RF} 下的结果并不理想。这是因为树上距离丢失掉了树的一些层次信息（例如父亲-孩子关系）。建立动态二分枝树的方法（BinaryDistance 与 BinaryOrder）的拟合度都较低，说明二分枝层次结构不能很好地拟合每个时间点的文档分布。这个结果和 Blundell 等人的结果[5] 一致。

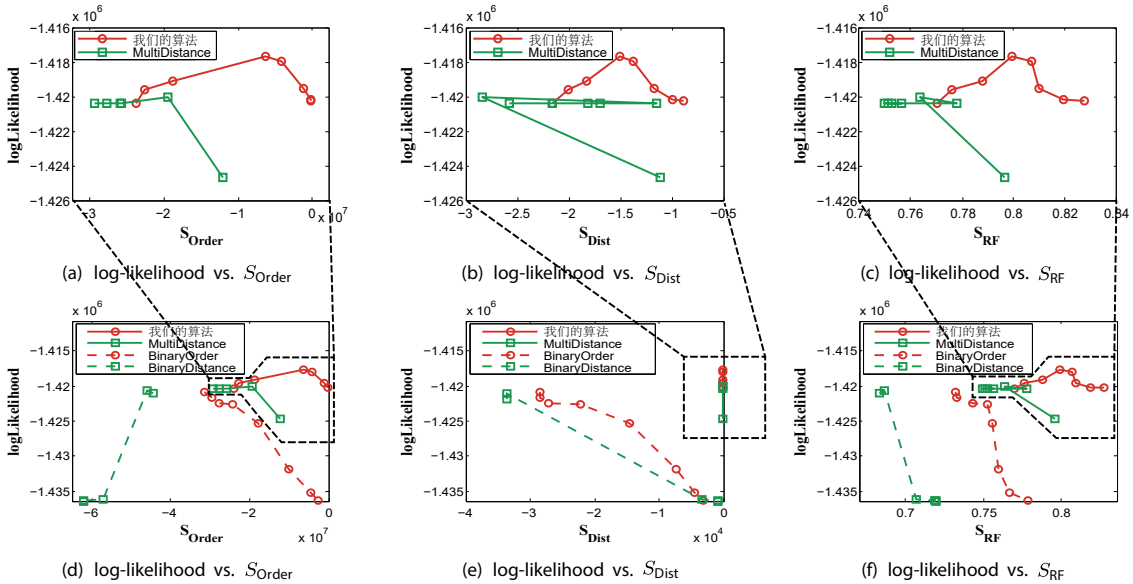
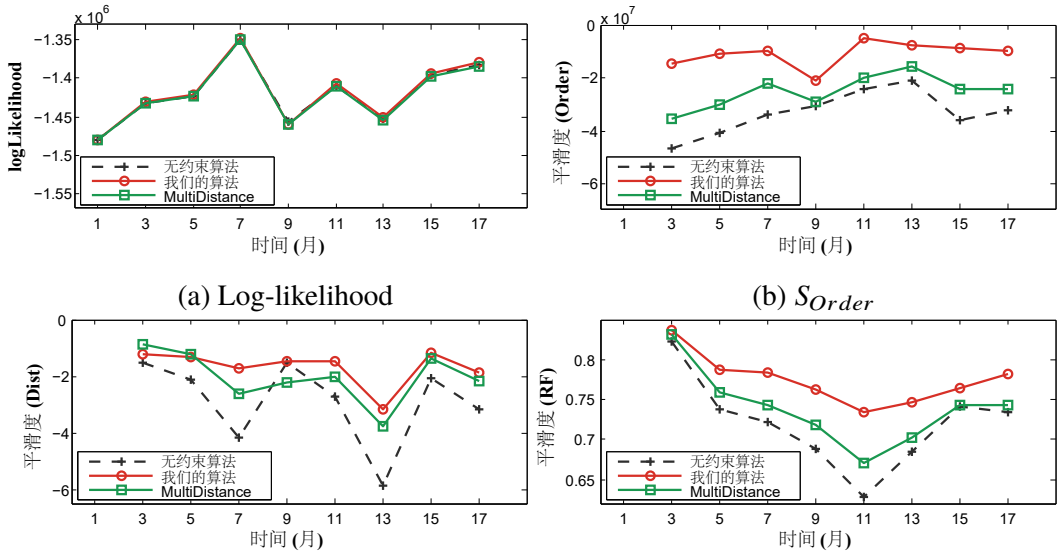


图 3.11　不同方法在不同约束项权重时的平滑度与拟合度对比。

另外，建立动态二分枝树的方法得到的平滑度也较低。这是因为建立动态二分枝树的方法更容易受到噪音的影响。通过检查结果我们发现，生成的二分枝树在不同的时间点改变往往较为剧烈。另外，BinaryOrder 相比于 BinaryDistance，生成的动态二分枝树的平滑度更高。这又一次说明了我们的基于三元组约束和扇形约束的约束模型比基于树上距离的约束模型效果更好。

第二，随着约束项权重不断增加，我们算法的平滑度也不断增加，似然概率则首先增加，然后减少。这说明加一定程度的约束实际对增加主题树的拟合度有好处。这是因为此时可靠度很大的三元组约束以及扇形约束被保留下来。这些三元组约束和扇形约束可以引导贪心的建树算法找到更好的解决方案。但是，MultiDistance 算法没有这种规律。即便约束项权重不断增加，这个算法得到动态多分枝树的平滑度也不一定会升高。这是因为基于树上距离的约束模型没有完整地考虑树的层次结构信息，并且也没有处理相互矛盾的约束。相似的，BinaryOrder 方法的平滑度随着约束项权重的增加不断升高，但是 BinaryDistance 的却没有这样的规律。这再一次说明了基于三元组约束和扇形约束的约束模型比基于树上距离的约束模型效果更好。

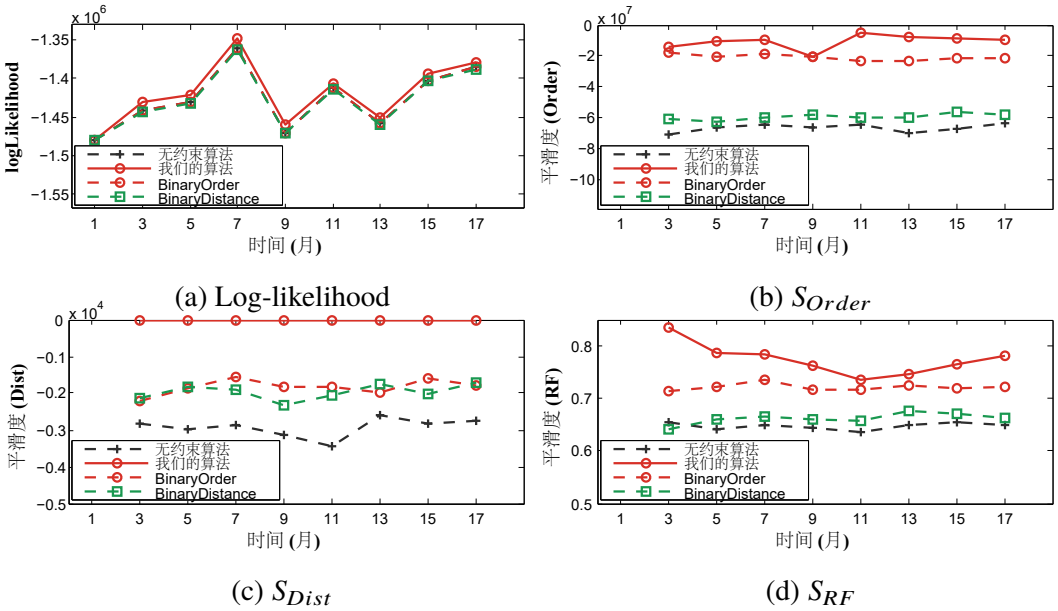
接下来，我们说明提出的算法在不同时间点都能较好地保证平滑度与拟合度。这个实验中，我们对八组不同约束项权重的结果进行平均，得到最终的结果。如图 3.12(b)、图 3.12(c)以及图 3.12(d)所示，我们的算法的平滑度几乎在每个时间点都优于 MultiDistance 算法的平滑度。另外，MultiDistance 方法的平滑度也比无约束算法的平滑度要好。这里，无约束算法相当于原始的贝叶斯多分枝树算法，它在每个时间点单独建立一棵主题树，没有考虑时间点之间的关联。在似然概率方面，



(a) Log-likelihood　　(b) S_{Order}

(c) S_{Dist}　　(d) S_{RF}

图 3.12　我们的算法和多分枝基准算法拟合度和平滑度随时间的变化情况对比。



(a) Log-likelihood　(b) S_{Order}

(c) S_{Dist}　(d) S_{RF}

图 3.13　我们的算法和二分枝基准算法拟合度和平滑度随时间的变化情况对比。

我们的算法、MultiDistance 算法以及无约束算法效果差不多（图 3.12(a)）。上面的实验结果再次说明我们的可以在不损失拟合度的情况下提高树的准确性。

进一步的，我们将提出的算法与建立动态二分枝树的基准算法进行比较。如图 3.13(b)、图 3.13(c)以及图 3.13(d)所示，我们的算法的平滑度几乎在每个事件点都优于 BinaryDistance 和 BinaryOrder 算法的平滑度。另外，BinaryDistance 和 BinaryOrder 的平滑度也比无约束算法平滑度要好。在似然概率方面，我们的算法比 BinaryDistance、BinaryOrder 和无约束的二分枝树算法的似然概率高（图 3.13(a)）。

3.3.4　算法效率

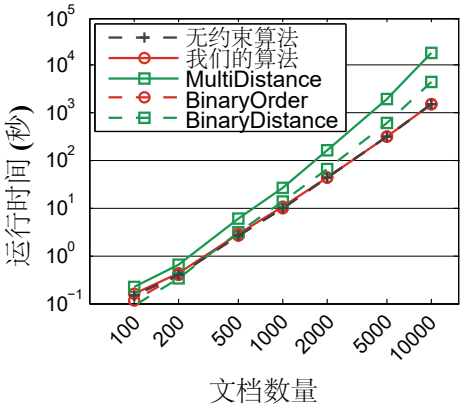
这一节中，我们首先证明提出算法的时间效率比 MultiDistance 和 BinaryDistance 高，同时跟 BindaryOrder 相当。然后，我们对比了提出的算法在不同的约束项权重时的效率。实验结果证明我们的动态多分枝算法与无约束的贝叶斯多分枝算法效率几乎一样好。

3.3.4.1　实验设置

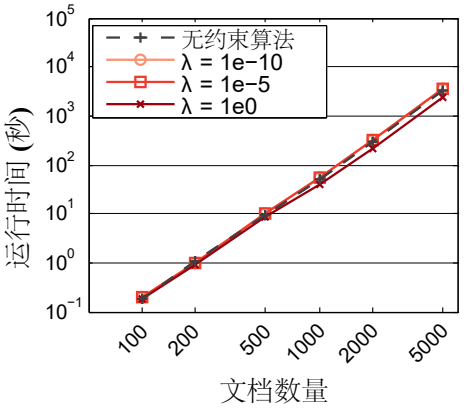
我们从纽约时报新闻中随机采样了 100,000 篇。在这个数据集上，我们测试了 EvoBRT，*K*NN-BRT 以及 SpillTree-BRT 的算法效率。我们将数据分为两组。我们首先利用第一组数据建立约束树，然后用第二组数据在约束树的基础上建立动态树。本实验中用到数据的词表长度为 665,261。

3.3.4.2　结果

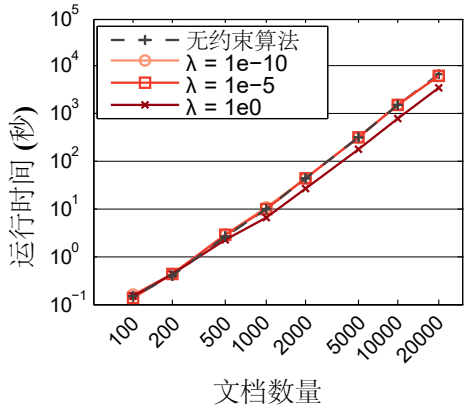
如图 3.14(a)所示，在基于 *K*NN-BRT 的情况下，我们的算法比 MultiDistance 的速度更快，时间效率更高。另外，我们算法的效率和无约束的 *K*NN-BRT 相当。图 3.14(b)、图 3.14(c)和图 3.14(d)分别显示了 EvoBRT、*K*NN-EvoBRT 以及 SpillTree-EvoBRT 在不同的约束项权重时的算法时间效率。实验结果再次说明我们的算法和无约束的 BRT、*K*NN-BRT 以及 SpillTree-BRT 算法运行时间相当。在这个例子中，当约束项权重较低的时候（例如为 1e-10 时），我们算法的运行时间和无约束算法的运行时间几乎一样。当约束项权重较大的时候，我们的算法在文本量较大时甚至比无约束算法更快。在检查了生成的树结构以后，我们发现带约束的算法此时约束树结构较为平衡。因此，生成的动态多分枝树结构也较为平衡。一般来说，BRT、*K*NN-BRT 以及 SpillTree-BRT 建立较为平衡的树时速度快很多。因为我们的算法时间复杂度较低，它可以在较短时间内处理较大规模的文档集合。



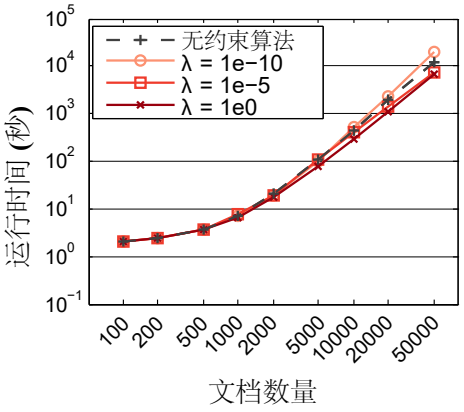
(a) 不同算法效率比较（*K*NN）



(b) EvoBRT 和无约束的 BRT 算法效率比较。



(c) *K*NN-EvoBRT 和无约束的 *K*NN-BRT 算法效率比较。



(d) SpillTree-EvoBRT 和无约束的 SpillTree-BRT 算法效率比较。

图 3.14　不同算法时间效率比较。

3.3.5　多约束树实验

在一些应用场景中，用户可能需要分析一棵已经存在了一段时间的子树如何随时间动态变化。例如，在 2013 年上半年，与 Windows 相关的主题、与 Xbox 相关的主题以及与销售和盈利相关的主题在几个月的时间里同时出现。一个微软的公关经理可能希望理解这三个主题之间的关联在 2013 年下半年是如何发展变化的，这样，她就可以渐渐理解公司在这些主要产品上的公关策略是否成功了。为了达到这个目的，我们需要保证不同的主题树上经常出现的子树结构更容易在后面的主题树种出现（平滑度）。在这个实验中，我们评估了基于多约束树的算法的有效性，并且研究了约束树棵数是如何影响树的平滑度的。本实验的实验设置和第 3.3.3节的平滑度与拟合度实验设置几乎一样。唯一的不同是我们不是只利用最后一棵树来建立约束树，而是利用最后 N_C 棵树来建立约束树。在这个实验中，我们测试了 $1 \leq N_C \leq 5$ 的情况。

3.3.5.1　评价标准

我们通过拓展第 3.3.3节中的平滑度指标，来衡量 T^t 与 T^{t-k} 之间的平滑度。

$$S^k_{Order} = \frac{1}{\lambda}\log(p_{Order}(T^t|T^{t-k})) \tag{3-22}$$

$$S^k_{Dist} = \frac{1}{\lambda}\log(p_{Dist}(T^t|T^{t-k})) \tag{3-23}$$

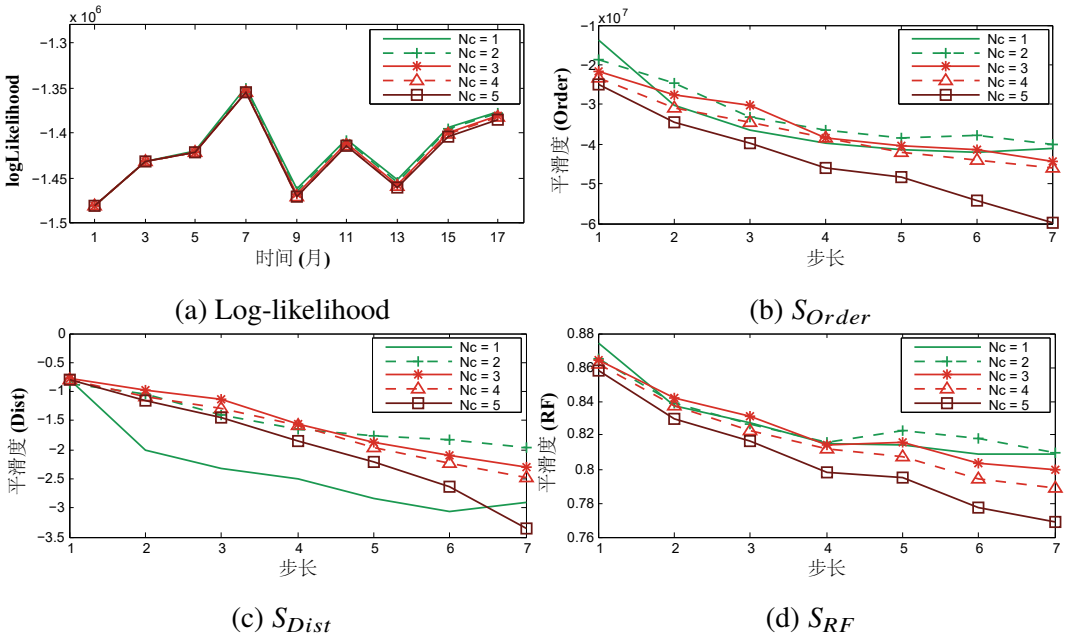
$$S^k_{RF} = 1 - \frac{d_{RF}(T^t, T^{t-k}(\mathcal{D}^t)) + d_{RF}(T^{t-k}, T^t(\mathcal{D}^{t-k}))}{2} \tag{3-24}$$

这里，k 是我们考虑的树之间平滑度的步长。在之前的实验中，我们只考虑了相邻两棵树之间的平滑度（$k = 1$）。在这个实验中，我们会考虑 $k > 1$ 的情况。这是为了验证我们的算法是否能够在较长时间内保持之前的子树结构。

3.3.5.2　结果

图 3.15(b)、图 3.15(c)与图 3.15(d)展示了算法平滑度如何随约束树个数变化。图中可以看出，约束树个数刚开始变多时，步长大于 1 的平滑度往往更高。但是，当约束树个数变得更多时，树的平滑度反而下降了。例如，在 $N_C = 5$ 时，平滑度在三种衡量标准下都有所下降。如图 3.16所示，这是因为太多的约束树会引入更多的相互矛盾的约束，从而导致树的平滑度降低。

接下来，我们证明，平滑度和拟合度都受到相互矛盾的约束的影响，从而说明我们提出的基于图匹配的方法可以提高平滑度与拟合度。图 3.17展示了树的似然概率以及平滑度如何随矛盾约束的比例变化。这里，我们测试了矛盾约束比例以



(a) Log-likelihood　(b) S_{Order}

(c) S_{Dist}　(d) S_{RF}

图 3.15　多棵约束树时算法的平滑度与拟合度。这里 N_c 代表约束树的个数。

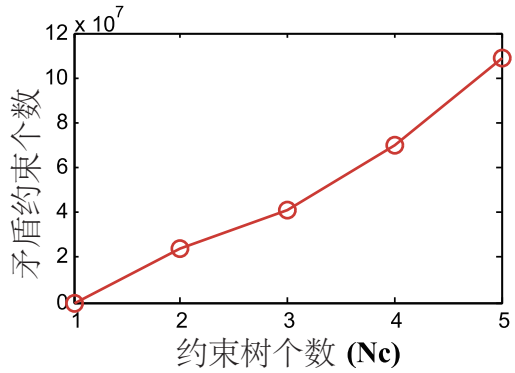


图 3.16　更多的约束树将导致更多矛盾约束的产生。

0.1 为步长，从 0 变成 1 的情况。图中可以看出，树的似然概率随着矛盾约束比例的增加不断减少。同时，在三种平滑度衡量方式下，树的平滑度都是随着矛盾约束的比例降低的。这说明矛盾约束会降低生成主题树的平滑度与拟合度。

3.4　案例分析

这一节中，我们通过两个案例分析来说明我们算法的有用性。

3.4.1　案例数据分析

首先，我们将我们的算法以及基准算法应用到在案例分析中用到的两个数据集上，对这些算法在不同评价指标上的性能进行对比。案例分析中用到的第一个数据集是微软数据集（图 3.1）。第二个数据集是欧债危机数据集。这个数据集中

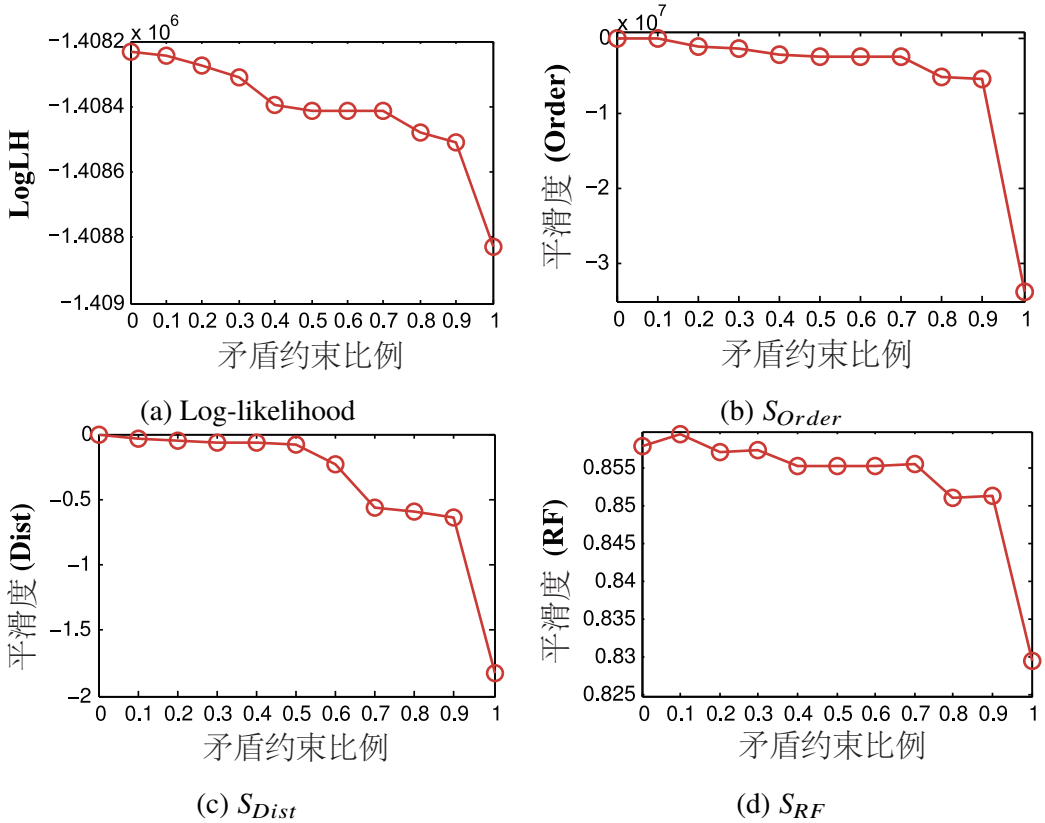


图 3.17　不同矛盾约束比例时的平滑度与拟合度。

表 3.5　微软数据上不同算法在不同评价指标上的结果。这里 n_d 代表树的平均深度，n_1 代表第一层的平均类别个数，n_2 代表树上平均中间节点个数。

	BinaryDistance	BinaryOrder	MultiDistance	我们的方法
似然概率	-987123.01	-965588.08	-952959.24	-916420.34
S_{Order}	-12052929.74	-465636.32	-2719459.86	-104211.85
S_{Dist}	-123.42	-116.45	-1.21	-0.72
S_{RF}	0.91759	0.910907	0.961792	0.964418
运行时间（秒）	479.51	478.07	977.19	651.71
n_d [min, max]	111 [38, 295]	117 [35, 362]	4 [3, 7]	4 [3, 5]
n_1	2	2	53	21
n_2	2375	2375	103	99

含有与欧债危机相关的 288,423 篇新闻，时间跨度从 2012 年 2 月 1 日到 2012 年 7 月 24 日。我们将数据按照周分为 25 组，每组中含有一周的数据，一共建立了 25 棵多分枝主题树。我们的方法生成的多分枝树的平均深度为 4，平均中间节点个数是 276，第一层的平均类别个数为 77。

这里，我们对提出算法的优势进行简单说明。对于我们的算法和所有基准算法，我们首先利用网格搜索找到使得似然概率最高的参数 γ 与 α。然后，我们再通

表 3.6　欧债危机数据上不同算法在不同评价指标上的结果。这里 n_d 代表树的平均深度，n_1 代表第一层的平均类别个数，n_2 代表树上平均中间节点个数。

	BinaryDistance	BinaryOrder	MultiDistance	我们的方法
似然概率	-4603529.8	-4572004.32	-4333273.41	-4292360.54
S_{Order}	-1099465913	-13693773.8	-49185577.18	-343917.64
S_{Dist}	-793.13	-709.88	-2.32	-0.24
S_{RF}	0.941171	0.947099	0.97298	0.977457
运行时间（秒）	9409.47	11709.53	44480.8	27213.76
n_d [min, max]	123 [73, 235]	215 [136, 281]	4 [3, 15]	4 [3, 7]
n_1	2	2	187	77
n_2	11536	11536	468	276

过网格搜索找到能够使得拟合度和平滑度比较平衡的约束项权重 λ。利用这些参数，我们在不同算法下建立了动态主题树。表 3.5和表 3.6显示了不同算法建立的动态主题树的性能。从表中可以看出，我们的算法在拟合度、平滑度方面都优于基准算法，这和数值实验部分的结论是一致的。

表 3.5和表 3.6有关树结构的性能指标，例如树的深度 n_d、树上第一层类别个数 n_1、树上中间节点个数 n_2 以及树的平滑度等，说明了我们的算法生成的树结构最平衡，随时间变化也最平滑。BinaryDistance 和 BinaryOrder 算法生成的主题树非常深，在两个数据集上都超过了 100 层。与此同时，它们在第一层的类别个数 n_1 均为 2，中间节点个数 n_2 至少有几千。这样的主题树结构并不准确，也很难被用户理解[5]。另外，BinaryDistance 和 BinaryOrder 生成的树结构在不同时间点上深度差异往往很大。这进一步加重了用户进行跟踪、理解的难度。例如，BinaryDistance 生成的与 Windows 相关的树结构在第一周的深度为 141，到了第二周却变成了 46。MultiDistance 生成的树结构过平，这体现在它生成的主题树第一层有特别多的节点（每个节点代表第一层的一个类）。在欧债危机数据集中，MultiDistance 第一层的平均类别个数为 187，这导致第一层有很多粒度过细的类别，甚至有一些重叠的类。例如，用我们的算法生成的结果中，第一层只有一个与经济相关的子树（图 3.20）。但是 MultiDistance 生成的结果中，相应时间点上第一层却有 6 棵不同的子树。这 6 棵子树中的主题分别为“shares, spain, rate”、“index, recession, april”、“oil, energy, crude”、“oil, crude, barrel”、“gold, fed, precious”以及“gold, price, bet”。通过检查相应的新闻，我们发现“oil, energy, crude”和“oil, crude, barrel”都在讨论石油交易，应该被合成一个主题。相似的，“gold, fed, precious”和“gold, price, bet”也应该被合成一个主题。另外，太多与经济相关的第一层主题将阻碍用户更好地追踪与经济相关的子树结构如何随时间变化。

3.4.2　微软数据集

第一个案例分析说明我们的算法如何帮助用户从多个角度（全局规律与局部细节）分析文档集合。我们在本章开头介绍了我们的算法如何分析层次结构的全局变化规律。在图 3.1(b)中，部分与 Xbox 相关的主题分裂出来，在第二周跟与 Windows 相关的主题进行了合并。为了了解为什么会发生这样的现象，我们仔细研究与 Xbox 相关的相应子树（T1）和与 Windows 相关的相应子树（T2）的层次结构。如图 3.18所示，与 Xbox 相关的子树第二层有两个类。其中一个与终极格斗冠军赛（Ultimate Fighting Championship, UFC）相关（**A**），另一个与 Xbox 的一些主要产品相关（**B**），包括“Xbox Live”（**B1**）、“Xbox 720”（**B2**）与“Xbox 360”（**B3**）。从表示对应关系的红边中，我们可以看出这两个主题中，是主题 **B** 与 Windows 相关主题产生了交互。为了了解具体是哪个产品与这次交互有关，我们检查了子主题 **B1**、**B2** 与 **B3**。我们发现，有两条表示对应关系的红边将这些子主题和与 Windows 相关的第三层的子主题相连。其中，第一条红边连接的是“Xbox Live”（**B1**）与“Windows phone”（**C1**）。通过浏览相应新闻，我们发现这是因为有人发起话题，讨论 Xbox Live 上的游戏（“Must Have Games”）是否应该重新在 Windows phone 中提供。一个相应新闻的标题为“‘Xbox LIVE Must Have Games’is back on Windows Phone”。另一条红边连接的是“Xbox 360 sales”（**B3**）与“Windows sales”（**C2**）。通过浏览相应新闻，我们发现这两个产品之所以相关，是因为在微软刚公布它第二个季度的盈利情况时，人们常常将 Xbox 销售的成功与 Windows 销售的滑坡作对比。一个相应新闻的标题为“Server and Xbox Overcome Windows Weakness in Microsoft’s 2Q”。

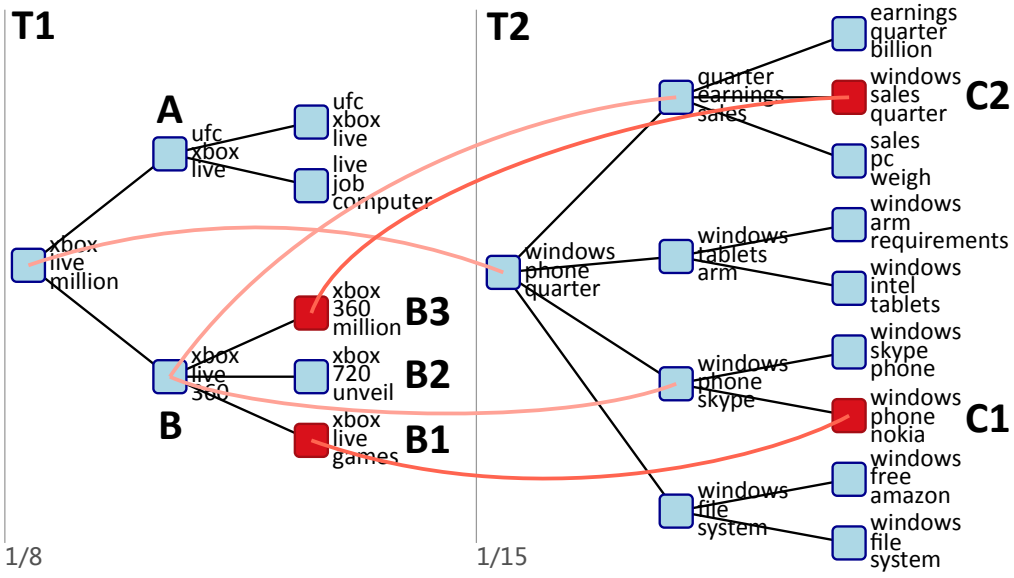


图 3.18　图 3.1中子树 T1 和 T2 的详细信息。相邻多分枝树的对应关系用红色曲线表示。

3.4.3　欧债危机数据集

第二个案例分析说明我们的算法如何帮助用户分析层次结构动态变化规律。

图 3.19利用 TextFlow 可视化技术[1] 展示了欧债危机数据上第一层的选中主题的主题内容及主题间关系如何随时间动态变化。图中主要有四个主题，它们分别是“fund”（红色）、“economy”（绿色）、“greece”（黄色）以及“china”（紫色）。根据与希腊相关的主题（“greece”）的幅度变化，我们可以将这段时间的欧债危机数据分为两个阶段。第一个阶段，与希腊相关的主题相对独立，与其他主题的交互不多；第二个阶段，这个主题与其他主题的交互明显变多了。为了了解造成这个现象的原因，我们仔细研究与希腊相关的主题。第一阶段中，希腊相关的主题主要讨论的是对希腊进行援助（“bailout”）。这些援助由其它欧洲国家提供给希腊，希望能够帮助希腊减少债务，从欧债危机的震中希腊入手来解决问题（“epicenter of the Eurozone crisis problem”）。在这个阶段中，不同的主题之间相对独立并且热度逐渐降低。这是因为援助希腊的国家在 2 月 15 日那一周就基本达成了共识，决定给希腊提供 1,700 亿美元的援助（“$170B Greek bailout approved”），不过一直到 3 月 14 日所在的那一周才正式作出决定（“Eurozone leaders formally approve Greek bailout”）。因为这段时间的新信息不多，因此主题不活跃，主题间交互也较少。第二个阶段中，希腊相关的主题主要讨论的是希腊的总统大选。5 月 6 日，希腊第一次大选没能建立政府。这次选举失败增加了希腊退出欧盟的危险，而希腊退出欧盟将对欧债危机造成严重影响（“Could the euro survive a Greek exit?”，“Greek exit would see turmoil ‘kick off’ again”）。因为问题的严重性，这段时间主题非常活跃，讨论热度变高，而且交互变频繁了。希腊相关的主题在希腊第二次大选（6 月 17 日）以后渐渐变得独立，而且热度变低了。这是因为支持希腊接受欧洲国家援助的政党赢得了选取并且成功建立了政府（“pro-bailout parties won enough seats to form a joint government”），因此对希腊退出欧盟的恐慌渐渐消退了（“fears of an imminent Greek exit ... receded”）。

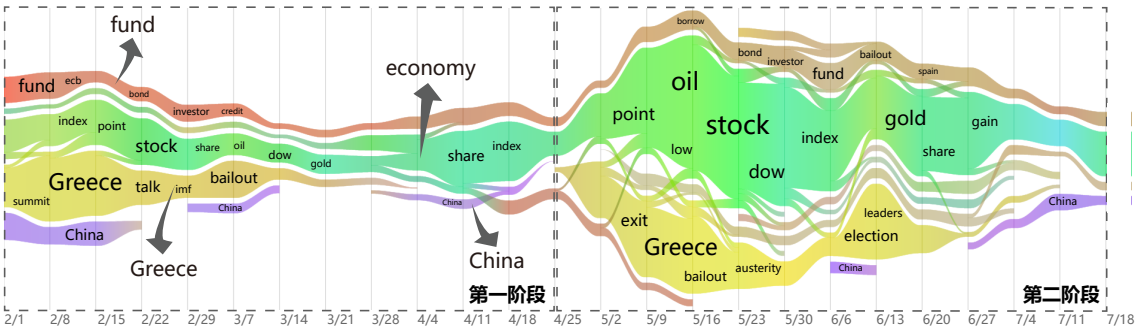


图 3.19　2 月 1 日至 7 月 24 日的欧债危机数据可视化结果。可以看出欧债危机被分为两个阶段。第一个阶段的主题之间相对独立，第二个的主题之间交互较为频繁。

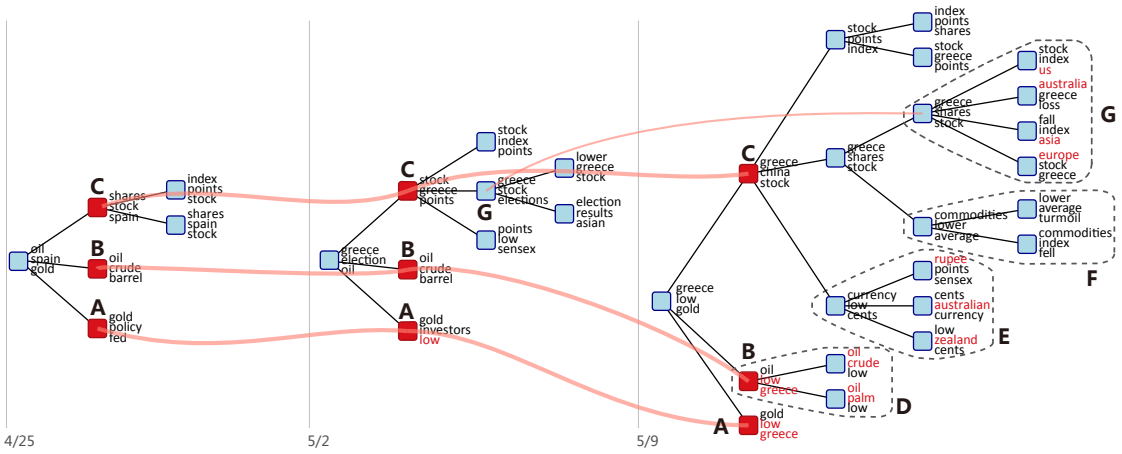


图 3.20　4 月 25 日至 5 月 15 日内，与经济（“economy”）相关子树的动态变化。因为希腊要退出欧盟引起了大量的讨论，子树的层次结构随时间不断增长。

因为希腊要退出欧盟是一个非常重要的事件，我们更仔细地对第二个阶段进行研究。具体来说，我们深入探索了与经济相关的主题（“economy”），因为它在这个阶段与希腊相关主题的交互最为频繁。图 3.20展示了经济相关主题从 4 月 24 日到 5 月 15 日期间的层次结构的变化。之所以选择这段时间，是因为这段时间希腊相关主题与经济相关主题的交互最为频繁。我们可以看到这段时间经济相关主题中有三个子主题：“gold”（**A**）、“oil”（**B**）以及“stock”（**C**）。我们发现从 5 月 6 日希腊第一次大选开始，这三个子主题都受到了希腊的影响。黄金（“gold”）相关子主题（**A**）的内容逐渐变化，主要的表现是关键词“low”和“greece”权重渐渐增大。这是因为希腊退出欧盟的危险造成了黄金价格的降低。例如，相关子主题中有一篇新闻的标题是 “Gold falls to 4-1/2 month **low** on **Greece** risks"。石油（“oil”）相关子主题（**B**）的内容变化与黄金相似。例如，石油相关子主题中有一篇新闻的标题是“Oil price at **lowest** of year: **Greece**, European debt crisis blamed”。另外，这个子主题结构也在增长。5 月 9 日那周，这个主题有了两个三层的子主题。其中一个是与棕榈油（Palm Oil）相关，另一个与原油（Crude Oil, **D**）。股票（“stock”）相关子主题（**C**）在这段时间增长非常迅速，从一棵两层的子树长成了一棵四层的子树。这期间，一个第三层的子主题“currency”（**E**）在股票相关子主题下出现。这个主题与货币相关，它主要讨论的是希腊退出欧盟的危险是如何影响其他国家的货币的。受影响的国家范围较广，包括新西兰（“NZ dollar falls as Greek woes heat up”）、澳大利亚（“Australian dollar drops 1%”）以及印度（“Rupee hits all time low of 54.46”）。除此之外，一个第四层的与日用商品相关的子主题“commodities”（**F**）也在股票相关子主题下出现了。这是因为“post-election turmoil in Europe drove down stocks and commodities; Dow ends down 76”。股票的子主题下，一个第三层的子主题 **G** 变到了第四层。与此同时，这个子主题的孩子节点个数还增加了。这表现出关

于希腊退出欧盟对股票市场造成影响的讨论在增加。图中，我们可以看出希腊退出欧盟影响了多个国家及地区的股票市场，包括欧洲（“European stocks fall sharply amid Greece woes”）、亚洲（“Asia stocks fall Amid political turmoil in Greece”）、澳大利亚（“Australian stocks end lower on Greek concerns”）以及美国（“U.S. stock-index futures decline on Greece’s political impasse”）。从上面的讨论，我们可以看出我们的方法能够很好地体现主题的内容和层次结构如何随时间变化。

3.5　小结及结论

在这一章，我们提出了自动挖掘多分枝主题树及其动态变化的算法 EvoBRT。我们通过一个贝叶斯在线滤波框架来对动态主题树进行建模。为了建立多分枝主题树，我们利用了目前最先进的建立多分枝主题树的方法——贝叶斯多分枝树。为了保证不同时间点的多分枝主题树之间的平滑度，我们利用一个条件先验概率来考虑之前时间点建立的多分枝主题树的结构。具体来说，我们引入了三元组约束和扇形约束的概念。这两个约束的优点是它们包含一棵树的所有层次信息。为了能够快速、有效地计算树之间层次结构的区别，我们定义了一个约束树来代表三元组约束和扇形约束集合，并且提出了约束树上的一系列操作，保证三元组约束和扇形约束的一致性。

实验结果表明，我们的算法在建树质量、平滑度、拟合度以及算法效率上均优于基准算法。算法时间复杂度分析说明我们的算法可以被应用到较大规模的文档集合。另外，两个真实新闻数据上的案例分析展示了我们的算法能够帮助用户从多层级研究文档集合中主题内容及主题间关系的动态变化。

第 4 章　多源静态文本的主题挖掘与可视分析

现实生活中，一个大事件（例如埃博拉）或者一些组织（例如 IT 公司）相关的主题往往分散在多个文本源，例如新闻、博客和微博。这些主题中，有些是多个文本源共有的主题，有些是单个文本源独有的主题。只有全面了解这些共有主题和独有主题，人们才能对这个大事件或者这些组织有个全面、完整的了解。近来，一些研究成果表明，对一个事件全面、完整的了解有助于人们做出更好的决策[69]。

但是，要对一个事件或者一些组织进行全面、完整的了解往往是一件费时费力的事情。人们需要在多个文本源中不停切换，才能渐渐对这个事件或者这些组织的情况产生清晰的认识。以一个微软的公共关系经理为例。为了对 NSA 窃听丑闻作出恰当回应，她需要了解新闻中该公司相关的主题以及这些主题之间的关联。另外，她还需要了解其他公司在相关主题上的回应。因此，她需要搜索、阅读其他公司的新闻，并与自己公司的新闻进行比较。在这个分析过程中，她需要在不同的文本集合间频繁切换，从而在脑海中形成对 NSA 窃听丑闻的清晰、完整的认识。

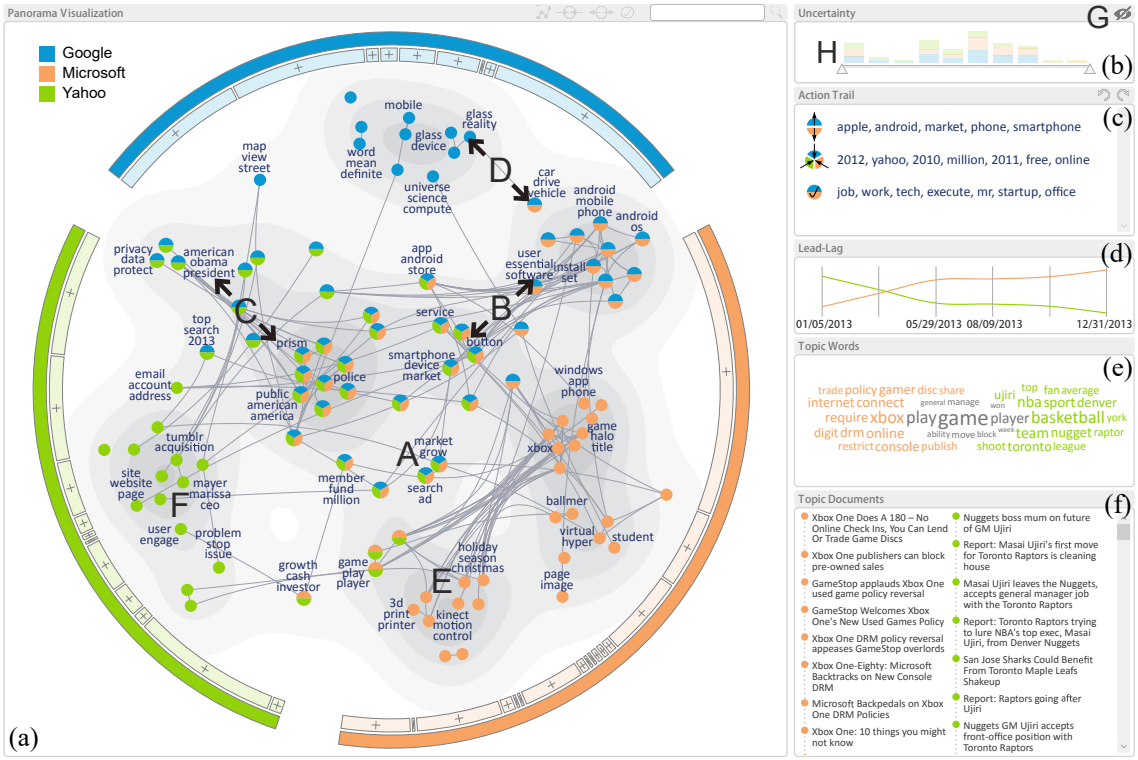


图 4.1　Google，Microsoft 以及 Yahoo 三家 IT 公司的主题全景图：(a) 用基于 LOD 的可视化技术分析探究匹配的主题图；(b) 不确定度筛选控件；(c) 匹配修改历史；(d) 领先-滞后分析；(e) 主题对应的词云；(f) 主题对应的文档。

从上面的例子我们可以看出，为了帮助人们更加方便、快速地了解一个事件或者一些组织，我们需要收集各个文本集合的信息，帮助他们在脑海中构建一副完整的画面。我们将这个完整画面对应的可视汇总（Visual Summary）称为主题全景图。

主题图（Topic Graph）对于主题全景图的构建有着重要的作用。主题图是一张节点-链接图。图中，每个点代表一个主题，每条边代表主题之间存在关联。通过主题图，用户可以快速、全面地分析感兴趣的主题[97,98]。因此，一个直接、自然的构建全景图的思路是将多个文本源的数据收集起来合并在一起，然后利用相关主题模型 CTM[97] 在合并后的数据上建立一张主题图。这个方法有两个问题。第一，不同的文本集合中，文本字符串的长度和语言使用习惯都有所不同。例如，新闻往往是比较正规的长文本，推特往往是噪音较大的短文本。因此，用同样的方法生成主题图难以对每个文本集合中的主题都进行很好的拟合。第二，即便在文本长度和和语言使用习惯较为相似的文本集合中，主题分布情况也往往不相同。直接用同样的参数对不同的文本集合进行建模，往往难以对不同文本集合的多样性进行很好的分析。我们在第 4.6.1节中对比了用一个模型对所有文档进行建模和用多个模型对各个文本集合的文档分别建模的结果。结果显示，后者的建模效果更好。这个实验结果也进一步论证了我们的观点。

为了更好地帮助用户对一个事件或者一些组织产生全面、完整的了解，我们开发了一个可视分析工具，TopicPanorama。图 4.1显示了用户利用 TopicPanorama 分析 Google，Microsoft 以及 Yahoo 三家 IT 公司的全景图时的结果。图 4.1(a) 中显示了这三家公司的一些共有主题和独有主题。例如，政府相关的主题一部分是三家公司共有的，一部分是 Google 和 Yahoo 共有的（**C**）。与 Kinect 相关的主题主要在 Microsoft 的新闻集中被提到（**E**）。通过这个概览，用户可以快速找到自己感兴趣的主题，对它们进行进一步探究。

从技术上说，TopicPanorama 将多张主题图一致地整合在一起，从而支持用户对主题全景图进行迭代式的分析。为了达到这个目的，我们进行了下面几项创新。

首先，我们开发了一个对多张图进行匹配的算法。算法可以将多张主题图进行一致的匹配，生成多个文本源的完整的主题图。我们的算法是基于应用广泛的图编辑距离方法[54] 设计的。我们算法的主要特征是可以对多张图进行一致性的匹配。现有方法在匹配多张图时，往往是直接对这些图两两之间进行匹配。这样的方法容易引入不一致的匹配结果。如图 4.3所示，对图两两之间进行匹配时，有可能将 G_1 中的 v_2 匹配到 G_2 中的 v_4。但是这与 G_1 中的 v_2 匹配到 G_3 中的 v_9 不一致了。我们的算法将对多张图的匹配结果进行统一优化，从而避免这种不一致的匹配。

然后，我们开发了一个增量式匹配修改算法，让用户可以修改匹配结果。我们

将度量学习与特征选择算法相结合，减轻用户在修改匹配结果时的负担。与之前的基于规则的方法[99] 不同，我们的算法在得到用户反馈以后，自动学习不同主题的匹配代价，更新匹配结果。除了用户修改的匹配，与用户修改的匹配相似的匹配也被修正，从而减轻用户修改负担。我们利用特征选择与增量式匈牙利算法进行加速，使得用户可以实时修改匹配结果。在三个真实数据集上的实验结果表明，我们的算法相比基于规则的算法，有效性至少提高了 20%。另外，我们的算法在所有实验中都能在少于 0.4 秒的时间内返回结果。

最后，我们开发了基于 LOD 的可视化技术，帮助用户理解匹配后的主题图。这个可视化将径向冰柱树与基于密度的图可视化进行结合，可以帮助用户从多个层级分析文档集合中的主题。我们利用一个带约束的弹簧模型进行布局，保证匹配图有较好的可读性与稳定性。另外，我们还设计了一系列交互操作，帮助用户从多个角度分析主题全景图。

4.1　任务分析与系统框架

我们通过采访领域专家，总结出 TopicPanorama 需要帮助用户完成的分析任务，并根据这些分析任务设计系统框架。

4.1.1　任务分析

六位专家参与了 TopicPanorama 的设计与讨论。其中，两位是 IT 公司公共关系经理（R1，R2），两位是记者（J1，J2），还有两位是媒体传播专业的教授（P1，P2）。这些专家平时是手动分析所有文本，从而对一个大事件或者一些组织进行全面的了解。他们表示，这个过程耗时耗力，并且需要较深的专业背景。他们希望有一个工具可以帮助他们分析更大的数据集，并且可以增进他们对感兴趣主题的理解。

在系统设计过程中，我们通过询问专家下面的问题，了解他们的分析需求。

- 你们是如何建立主题全景图的？
- 你们希望如何对全景图进行探索与分析？
- 在你们平时的工作中，一般是如何应用主题全景图的？

通过对专家采访的总结分析，我们整理出来了一下几个抽象的分析任务。

T1 - 获得相关主题的概览。 所有专家都表示，在对全景图进行分析时，希望能看到一个含有所有主题的概览。他们说一个可以有效分析两至三个文本源的工具将对他们的工作有较大帮助。这和以前的论文中的结论相符。论文表示，人在进行可视化比较时，大约可以跟踪四个物体[100]。

T2 - 分析不同文本源的共有主题与独有主题。 在分析一个主题图时，专家们往往

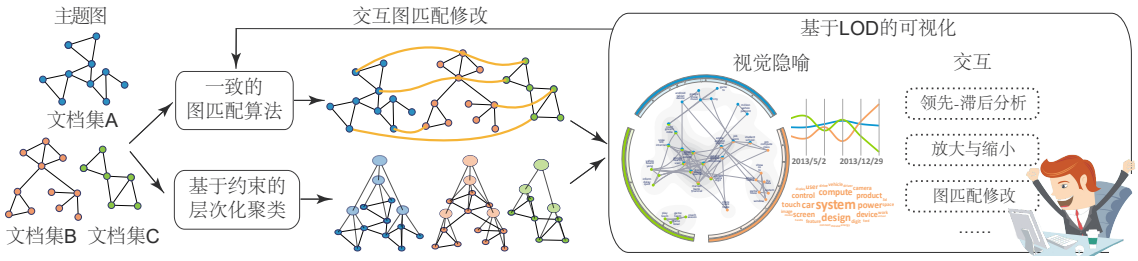


图 4.2　TopicPanorama 系统框架概览。

希望比较不同文本源的主题有什么共同点和区别。具体来说，他们主要希望了解不同文本源的共有主题和独有主题。他们还希望能在一个可视化界面上同时进行共有主题和独有主题的分析。

T3 - 分析主题之间的关联。 所有的专家都希望分析主题之间的关联。他们特别关注不同文本源中共有主题和独有主题之间的关联。他们表示，这样能帮助他们更好地找到有趣的信息。专家 P1 提到：“在分析媒体的框架理论时，我需要了解两个看似不相关的议题（例如大众媒体的议题和草根议题）是如何互相影响的。”

T4 - 从多个主题层级分析主题全景图。 在很多应用中，一个文本源可能含有几百上千个主题。在很多任务中，用户都需要快速地了解这些主题的概览，然后逐渐发现感兴趣的主题，仔细研究这些主题的具体相关内容。例如，R1 说到：“在我日常工作中分析的多个文本源往往含有几千个主题。一个能够帮助我有效组织这些主题的工具将对我有很大的帮助。”

T5 - 分析匹配主题的随时间变化的规律。 专家表示，他们在研究感兴趣的主题时，往往希望研究主题在多个文本源如何传播。在我们的应用中，三个专家表示希望看到在匹配的主题上，不同文本源的领先-滞后关系如何随时间变化。例如，记者 J1 说到：“在对不同媒体传播事件的规律进行比较时，能够知道哪个媒体处于领先地位往往很有用。它能帮助我们分析哪个媒体首先播报了这个新闻。”

T6 - 根据用户需求定制主题全景图。 在很多现实应用中，给定一些文本源，不同专家在完成不同任务时，需要的主题全景图也往往不尽相同。因此，他们需要能够根据自己的信息需求定制全景图。例如，在分析 IT 公司相关的主题时，R1 更加关心与游戏相关的主题，因此她希望能将这里面匹配不正确的主题进行修正。

4.1.2　系统框架

为了帮助用户更好地完成上述任务，TopicPanorama 包含下面的主要功能：

- 利用一个主题图代表一个文本源，并且通过层次化聚类有效组织主题图中数目众多的主题（**T3**, **T4**）；
- 能够将不同主题图进行匹配，从而生成主题全景图（**T1**）；

- 一个基于 LOD 的视图，它可以将全景图中共有部分放在视图中间，独有部分放在视图中与相应文本源相近的部分（**T2**）；
- 丰富的交互，包括领先-滞后分析与交互式图匹配修改等（**T5**, **T6**）。

相应的，TopicPanorama 包含了四个主要模块：图匹配算法、层次化聚类、可视化模块以及交互模块（图 4.2）。给定一些主题图，图匹配模块生成一个一致的图匹配结果。为了有效组织含有大量主题的主题图，我们用一个带约束的贝叶斯层次化聚类算法[2] 来将主题图建立成多分枝主题树。这些主题树和图匹配结果一起，被送入可视化模块进行显示。可视化模块生成的视图中主要包含一个径向冰柱树和一个基于密度的图可视化，这两个图用于展示图匹配的结果。用户可以与这个图匹配的结果进行交互，进行进一步的分析探索。例如，用户可以修改一个主题的匹配结果，然后 TopicPanorama 会增量式地更新其他主题的匹配结果。

图 4.1显示了 TopicPanorama 的用户界面。这个用户界面主要包含两个区域。第一个区域（图 4.1(a)）是 TopicPanorama 主要视图。这个视图显示了多个文本集合所有相关主题的全景图，它主要包括径向冰柱树和基于密度的图可视化。第二个部分是一个信息面板，它用来显示一些主题和匹配的具体信息，包括显示匹配不确定程度的控件（图 4.1(b)）、匹配修改的历史（图 4.1(c)）、不同文本源在特定匹配主题上的领先-滞后关系（图 4.1(d)）、主题对应的词云（图 4.1(e)）以及主题对应的文档（图 4.1(f)）。

4.2　一致的图匹配算法

这一节中，我们主要介绍如何对多张主题图进行图匹配，找到主题之间一致的对应关系。

4.2.1　算法模型

图编辑距离法[54,56] 是一种被广泛应用的对两张图进行匹配的算法。它通过将一张图修改成另一张图所需要进行的修改步数来衡量两张图之间结构上的差异性。一般来说，常用的图修改操作是点的插入、删除以及替换。

给定两张图 $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ 以及 $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$，$\mathcal{V}_1$ 和 $\mathcal{V}_2$ 分别代表这两张图的点的集合，$\mathcal{E}_1$ 和 $\mathcal{E}_2$ 代表这两张图的边的集合。我们将它们之间的图匹配结果记为 $f_{G_1G_2}$。G_1 和 G_2 之间的图编辑距离定义为将 G_1 变为 G_2 的最小编辑代价：

$$d(G_1, G_2) = \min\ c(f_{G_1G_2}), \qquad c(f_{G_1G_2}) = \sum_{o_i} c(o_i) \tag{4-1}$$

这里，$c(f_{G_1G_2})$ 是匹配 G_1 和 G_2 的代价，$c(o_i)$ 代表编辑操作 o_i 对应的编辑代价。

给定 N 张图，要将一个针对两张图的匹配算法拓展成多张图的算法，一个自然的想法是直接对多张图两两之间进行匹配，将匹配结果合并为最终结果（基准算法一），即：

$$d(G_1, G_2, ..., G_N) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} d(G_i, G_j) \tag{4-2}$$

这个算法的问题在于它可能引入不一致的匹配。图 4.3展示了一个例子。图中所示的三张主题图是通过将 CTM 算法[10,97] 应用到三家 IT 公司 Yahoo（G_1）、Microsoft（G_2）和 Google（G_3）生成的。CTM 通过加入逻辑正态（Logistic-Normal) 先验，可以有效地从文本中提取主题以及主题之间关联[10]。在图中，相同颜色的节点是由专家标注的关于同一个主题的节点。例如，蓝色节点代表是关于印度法庭对 Google 和 Facebook 等公司进行起诉的主题，紫色节点代表的是关于 2012 年美国政府选举的主题。这里，$f_{G_1G_2}$ 将 v_2 匹配到 v_4。$f_{G_2G_3}$ 将两个紫色节点匹配在一起（$v_4 \mapsto v_7$）。这里，$v_i \mapsto v_j$ 表示节点 v_i 被匹配到了节点 v_j。通过这两个匹配，我们可以看出 v_2 被匹配到了 v_7。但是这个结果和 $f_{G_1G_3}$ 的直接匹配结果（$v_2 \mapsto v_9$）相矛盾。节点 v_3、v_5、v_8 以及 v_{10} 之间也有类似的不一致性。

一个简单的解决匹配不一致的方案是去除掉造成不一致的节点。但是，上述方法可能造成众多的不一致匹配。当造成不一致的节点数目众多时，我们可能很难找到一个最优的去除不一致节点的策略。另一个选择是将部分两两图匹配结果作为约束，用来推算其他图的匹配结果，从而消除不一致性（基准算法二）。这个方法可以保证所有图的共有部分的一致性。但是，它可能会造成非共有部分匹配的丢失。图 4.4展示了这种方法在三张主题图上进行匹配的结果。尽管匹配结果是

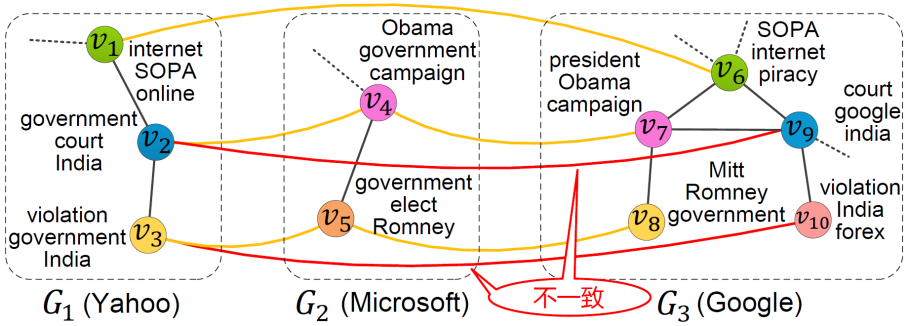


图 4.3　直接对多张主题图两两之间进行匹配造成的不一致匹配。G_1 和 G_3 之间的匹配结果是直接用两两之间的图匹配算法计算得到的，这两个匹配结果可以用 $f_{G_1G_2}$ 和 $f_{G_2G_3}$ 表示。它们之间的匹配结果显示 $v_2 \mapsto v_4$，$v_4 \mapsto v_7$。这个结果和 G_1 和 G_3 之间直接的匹配结果，即 $v_2 \mapsto v_9$ 不一致。这里 $v_i \mapsto v_j$ 代表 v_i 被匹配到主题 v_j。

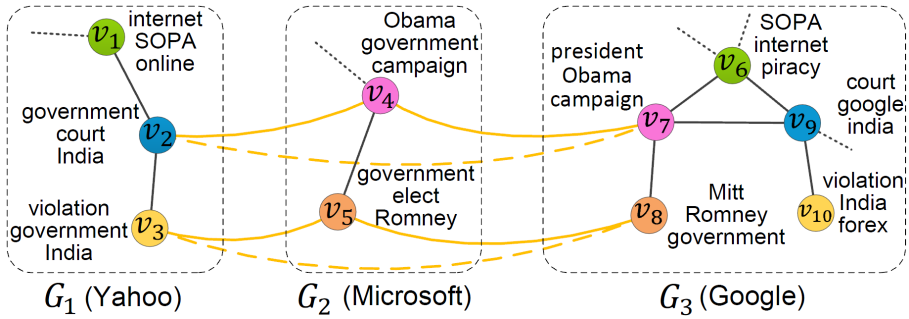


图 4.4　一个在多张主题图两两匹配结果上加入简单约束的例子。在计算 $f_{G_1G_2}$ 以及 $f_{G_2G_3}$ 时，$f_{G_1G_3}$ 被当作了约束。本应该匹配在一起的 v_1 和 v_6 没有对应关系。这是因为 v_1 和 v_6 在 G_2 中没有对应的节点。

一致的，但是算法丢失了一些在 G_2 中没有对应节点的匹配结果。例如，我们不知道 v_1 是否被匹配到了 v_6、v_9，因为 v_1、v_6 以及 v_9 在 G_2 中都没有对应的节点。

为了解决这个问题，我们开发了一个一致的图匹配算法。该算法在限定所有匹配结果是可传递（Transitive）的情况下，最小化所有图的匹配总代价。通过保证匹配是可传递的，我们保证了匹配结果的全局一致性。数学上，我们提出的图匹配算法如下：

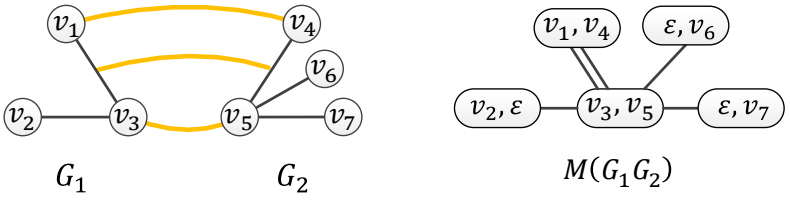
$$\begin{aligned} &d(G_1, G_2, ..., G_N) = \min\ c(f_{G_1G_2...G_N}), \quad c(f_{G_1G_2...G_N}) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} c(f_{G_iG_j}) \\ &s.t.\ v_l \mapsto v_m, v_m \mapsto v_n \Rightarrow v_l \mapsto v_n \\ &\forall G_i, G_j, G_k \in \{G_1, G_2, ..., G_N\},\ \forall v_l \in \mathcal{V}_i, \forall v_m \in \mathcal{V}_j, \forall v_n \in \mathcal{V}_k \end{aligned} \tag{4-3}$$

然后，我们将公式 (4-3) 中的代价函数重写为下面的形式：

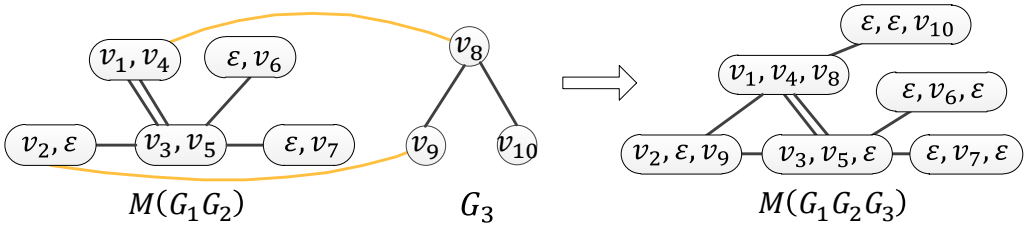
$$c(f_{G_1G_2...G_N}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N-1} c(f_{G_iG_j}) + \sum_{i=1}^{N-1} c(f_{G_iG_N}) = c(f_{G_1G_2...G_{N-1}}) + \sum_{i=1}^{N-1} c(f_{G_iG_N}) \tag{4-4}$$

为了进一步简化代价函数，我们提出元图的概念。元图是通过将匹配到一起的节点和边合并成元节点和元边而生成的。元图中是 N 张图进行一致性匹配的结果，它既包含多个文本源的共有主题，又包含单个文本源的独有主题。图 4.5(a) 显示了匹配结果 $f_{G_1G_2}$ 对应的元图 $M(G_1G_2)$。每个元图可以和普通图进行匹配，生成一张新的元图。此时，我们将每个编辑操作的代价定义为对元节点中包含的所有节点进行编辑的代价的总和。这样，对元图和普通图进行匹配的代价可以写为

$$c(f_{M(G_1...G_{N-1})G_N}) = \sum_{i=1}^{N-1} c(f_{G_iG_N}) \tag{4-5}$$



(a) $f_{G_1G_2}$ 和它对应的元图。



(b) 利用 $M(G_1G_2)$ 计算 $M(G_1G_2G_3)$。

图 4.5　一个元图以及在此基础上迭代进行匹配优化的例子。这里的 ϵ 代表一个空节点（Null Node）。

此时，公式 (4-4) 可以重写为

$$c(f_{G_1G_2\ldots G_N}) = c(f_{G_1G_2\ldots G_{N-1}}) + c(f_{M(G_1G_2\ldots G_{N-1})G_N}) \tag{4-6}$$

通过上面表达式，我们发现可以通过 $N-1$ 张图匹配结果得到的元图 $M(G_1G_2\ldots G_{N-1})$ 生成 N 张图匹配结果（$f_{G_1G_2\ldots G_N}$）的元图 $M(G_1G_2\ldots G_N)$。图 4.5(b) 显示了我们利用 $M(G_1G_2)$ 建立 $M(G_1G_2G_3)$ 的一个例子。

4.2.2　算法流程

直接优化公式 (4-6) 是 NP 问题。因此，我们用一个贪心迭代法寻找一个较好的近似解。对于 $\forall k, 2 < k \leq N$，我们首先匹配 G_k 和只考虑 $k-1$ 张图时最好匹配结果对应的元图，从而生成一个初始的图匹配结果 $f_{G_1G_2\ldots G_k}$。然后，我们对这个初始的匹配结果进行迭代优化。对于 $1 \leq i < k$，我们将匹配结果 $f_{G_1\cdots G_{i-1}G_{i+1}\cdots G_k}$ 固定，并且生成它的元图。然后，我们将这个元图与 G_i 进行匹配。如果新的匹配结果代价比之前的匹配结果低，我们就用新的匹配结果替代原来的结果。

我们用一个在三张主题图上进行匹配的简单例子说明算法的主要思想。图 4.6(a) 显示了初始匹配结果。与基准算法二（图 4.4）不同，我们的算法可以找到 v_1 和 v_6 之间的对应关系。这个初始结果并不是最优的，因为在 G_1 与 G_2 进行匹配时，我们并没有考虑到 G_3。因此，蓝色节点 v_2 被错误地匹配到了紫色节点 v_4。这个错误的匹配结果可能会在之后的匹配过程中造成更多的错误。为了解决这个问题，我们对初始的匹配结果进行了迭代式的优化。图 4.6(b) 展示了第一次迭代

Algorithm 1: 一致的图匹配算法。

Data: N 张主题图 $G_1, G_2, ..., G_N$

Result: 一致的匹配结果 $f_{G_1G_2...G_N}$

begin

$f_{G_1G_2} \leftarrow PairwiseMatch(G_1, G_2)$, $M^0 \leftarrow M(G_1G_2)$

for $k = 3 \rightarrow N$ **do**

$f^0_{G_1G_2...G_k} \leftarrow PairwiseMatch(M^0, G_k)$

$c_0 \leftarrow 0$, $c_1 \leftarrow c(f^0_{G_1G_2...G_k})$, $iter \leftarrow 0$

while $c_0 \neq c_1$ & $iter < MaxIter$ **do**

for $i = 1 \rightarrow k$ **do**

$f_{G_1G_2...G_k} \leftarrow PairwiseMatch(M^{k-1}(G_1...G_{i-1}G_{i+1}...G_k), G_i)$

if $c(f_{G_1G_2...G_k}) < c(f^{k-1}_{G_1G_2...G_k})$ **then**

$f^k_{G_1G_2...G_k} \leftarrow f_{G_1G_2...G_k}$

$f^0_{G_1G_2...G_k} \leftarrow f^k_{G_1G_2...G_k}$

$c_0 \leftarrow c_1$, $c_1 \leftarrow c(f^0_{G_1G_2...G_k})$, $iter \leftarrow iter + 1$

$M^0 \leftarrow M^k(G_1G_2...G_k)$

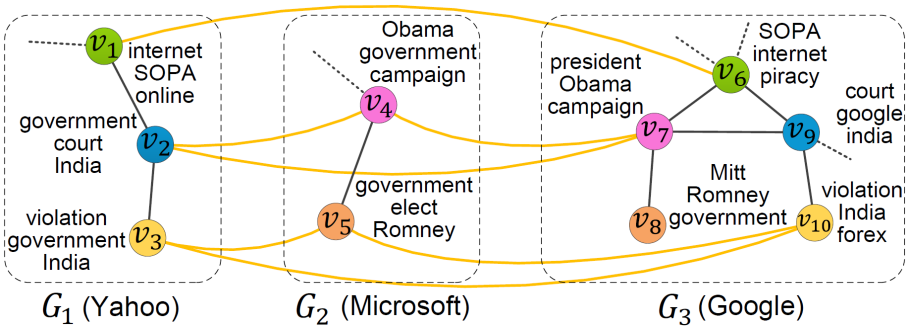
以后的结果。在这一步中了，G_2 和 G_3 之间的匹配固定成了一个元图。G_1、G_2 之间的匹配结果以及 G_1 和 G_3 之间的匹配结果被更新了。图中，粗的曲线代表固定的匹配，细的曲线代表更新的匹配。因为 $v_4 \mapsto v_7$ 之间的匹配固定了，$f^1_{G_1G_2G_3}$ 利用了这个信息，将 v_2 和 v_9 成功匹配到了一起。图 4.6(c) 显示了第二次迭代以后产生的最终匹配结果 $f^2_{G_1G_2G_3}$。在 v_3 被匹配到 v_{10} 的情况下，$f^2_{G_1G_2G_3}$ 成功地将 v_5 和 v_8 匹配到了一起。

4.3　交互式图匹配结果修改

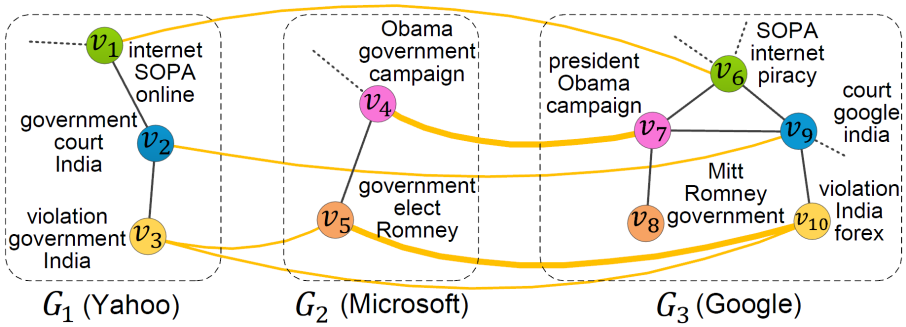
尽管我们提出的图匹配算法能够成功地生成多张图较优的匹配结果，这个结果可能还有不完美之处。另外，不同的用户有不同的信息需求。因此，一个图匹配模型可能难以满足不同用户的所有需求。为了解决这个问题，TopicPanorama 允许用户对匹配结果进行交互修改。

4.3.1　问题描述

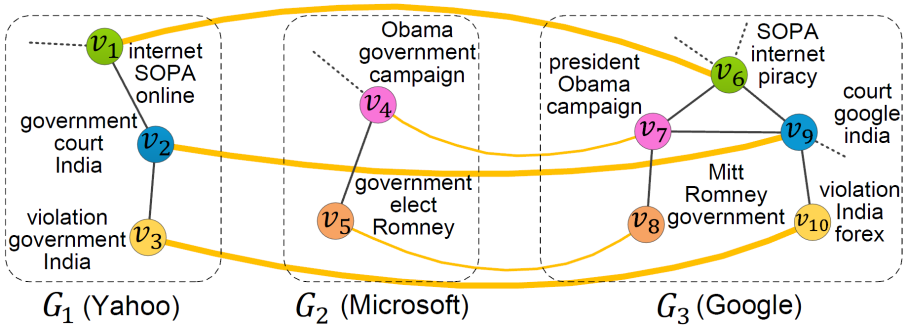
在 TopicPanorama 中，用户可以根据自己的知识以及信息需求，对匹配结果进行修改。我们支持两种用户反馈：两个主题应该进行匹配以及两个主题不应该进



(a) 初始匹配结果 $f^0_{G_1G_2G_3}$.



(b) 第一次迭代后，算法将 v_2 匹配到了 v_9。



(c) 第二次迭代后算法生成的最终匹配结果。v_5 被匹配到了 v_8。

图 4.6　一致的图匹配算法示例。

行匹配。我们将这个反馈结果看作是用户提供的约束，对图匹配结果进行优化。这些用户反馈可以看作图匹配算法中的一种编辑操作：节点替换。因此，我们可以用用户反馈修改节点替换代价，从而更新匹配结果。具体来说，我们的图匹配结果修改方法包含下面两步：

- 根据用户提供的约束修改节点替换代价。
- 根据更新了的节点替换代价，利用增量式匈牙利算法[101] 更新匹配结果。

第二步较为简单，主要用到的是现有的增量式匈牙利算法。我们这里主要介绍第一步。现有最先进的算法[99] 利用基于规则的方法修改节点替换代价。具体来说，这个方法将所有用户认为应该进行匹配的节点的替换代价设为 0，将所有用户认为不应该进行匹配的节点的替换代价设为正无穷。其他节点之间替换代价不变。这

个方法在两个主题共有的词重要性比较高时结果较好。但是，因为这个方法认为所有词的重要性一样高，当匹配的主题中的共有词的重要性较低时，算法受噪声影响较大，可能造成错误的匹配结果。在现实应用中，人们心目中不同词的重要性往往是不同的，而且不同词之间可能存在一定的相关性。例如，“xbox”和“playstation”是两个不同的词，但是它们都是关于视频游戏机的，因此它们有一定相关性。

为了解决这个问题，我们利用度量学习来自动学习节点的替换代价。我们还利用特征选择算法对度量学习进行加速，同时保证算法的有效性没有损失（见表 4.4）。

4.3.2　度量学习

用户给出一组彼此相似的样本时，度量学习通过约束这些样本之间的距离小于特定值来学习样本距离函数[102]。它通过修改一些特征的重要性以及特征之间的相关程度，来满足尽可能多的用户给定约束（即相似样本对）。具体来说，如果两个特征总是频繁出现在用户认为相似的样本对中，算法认为它们的关联性较强。在我们的应用中，需要学习的距离是节点替换代价。约束是一组用户指定的相似（匹配）主题和不相似（不匹配）主题。节点的替换代价根据平方马氏距离（Mahalanobis Distance）定义：

$$c(v_i \mapsto v_j) = d_A(\mathbf{w}_i, \mathbf{w}_j) = (\mathbf{w}_i - \mathbf{w}_j)^T A(\mathbf{w}_i - \mathbf{w}_j) \tag{4-7}$$

这里，A 是需要学习的正定矩阵。A 的对角线元素中存储的是词的重要性，非对角线元素中存储的是不同词之间的相关程度。$\mathbf{w}_i = (w_i^1, ..., w_i^n) \in \mathcal{R}^n$ 是主题 v_i 的词的分布。这里，n 是词表长度，w_i^k 是第 k 个词出现在主题 v_i 中的概率.

学习节点替换代价等价于学习词重要性与相关性的矩阵 A。要学习 A，有两个主要的挑战。第一个挑战是用户给定的约束往往较少（少于 50 个）。因为训练数据很少，生成的模型可能准确性不够。第二个挑战是用户提供的约束是一个一个到来的，因此我们需要增量式地更新 A。为了解决这两个问题，我们采用了一个基于信息理论的在线度量学习方法[102]。这个算法的主要目标是最小化损失，即最小化更新 A 以后，v_i 和 v_j 之间的实际距离 $d_A(\mathbf{w}_i, \mathbf{w}_j)$ 和用户心中理想距离 d_t 之间的差异：

$$A_{t+1} = \underset{A>0}{arg\,min} \left[D(A, A_t) + \eta_t \left(d_t - d_A(\mathbf{w}_i, \mathbf{w}_j) \right)^2 \right] \tag{4-8}$$

这里 $A \succ 0$ 表示 A 是一个正定矩阵。公式中第一项是一个正则项，它保证更新后的 A 矩阵和现有矩阵 A_t 比较相近。在线学习算法中往往有这一项，保证更新后的 A 能保持原来较优的特性。公式中第二项是损失项，它保证更新后的 A 使得 v_i

和 v_j 之间的距离和用户心目中的理想距离 d_t 较为接近。$\eta_t > 0$，是一个用于平衡正则项和损失项的参数。$D(A, A_t)$ 衡量的是 A 和 A_t 的不相似程度。

为了减少用户的负担，我们不要求用户输入理想距离 d_t。相反，我们根据用户的反馈（匹配或者不匹配），自动计算理想距离 d_t。对于被用户规定为匹配的主题，它们之间的理想距离应该比此时的真实距离 $d_{A_t}(\mathbf{w}_i, \mathbf{w}_j)$ 小。对于被用户规定为不匹配的主题，它们之间的理想距离应该比此时的真实距离 $d_{A_t}(\mathbf{w}_i, \mathbf{w}_j)$ 大。基于这样的发现，我们将理想距离定义为 $d_t = \alpha d_{A_t}(\mathbf{w}_i, \mathbf{w}_j)$。当用户规定主题进行匹配时，$0 < \alpha < 1$。当用户规定主题不匹配时，$\alpha > 1$。在实验中，我们用网格搜索的方式确定 α。具体来说，我们认为使得用户修改次数最小的 α 为最优值。

根据 Jain 等人的推理[102]，存在使得公式 (4-8) 最小化的解析解，该解为

$$A_{t+1} = A_t - \left(\eta_t(\bar{d}_t - d_t)A_t\Delta\mathbf{w}_t\Delta\mathbf{w}_t^T A_t\right) / \left(1 + \eta_t(\bar{d}_t - d_t)\hat{d}_t\right) \tag{4-9}$$

这里，$\Delta\mathbf{w}_t = \mathbf{w}_i - \mathbf{w}_j$，$\hat{d}_t = \Delta\mathbf{w}_t^T A_t \Delta\mathbf{w}_t$，而且

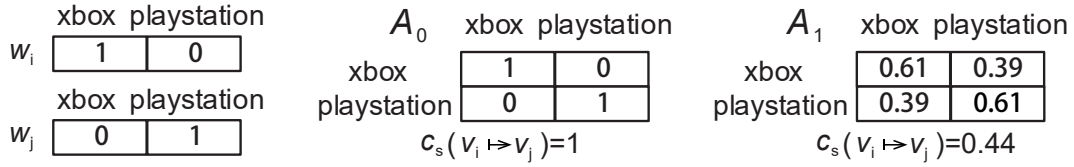
$$\bar{d}_t = \left(\eta_t d_t\hat{d}_t - 1 + \sqrt{(\eta_t d_t\hat{d}_t - 1)^2 + 4\eta_t\hat{d}_t^2}\right) / \left(2\eta_t\hat{d}_t\right) \tag{4-10}$$

一个应用度量学习的例子见图 4.7。一个用户对含有“xbox”这个词的主题与含有“playstation”这个词的主题进行匹配以后，我们的算法学习出来了“xbox”与“playstation”之间的相关性，使得相关性从 0 上升到了 0.39。相应地，这两个主题之间的替换代价变小了，使得这两个主题更容易进行匹配了。

4.3.3　特征选择

如果文本集合中所有的词（特征）都被用于度量学习中，算法的速度将会无法保证实时性。为了在保证算法有效性的基础上提高算法效率，我们利用特征选择挑选出最重要的特征，只将这些重要特征送入度量学习算法中。

现有的特征选择算法可以被大致分为三类：筛选法（Filter）、包装法（Wrapper）以及嵌入法（Embedded）[103]。这三种方法中，筛选法是效率最高的。为了尽可能



(a) 将 $\mathbf{v}_i$ 与 $\mathbf{v}_j$ 进行匹配　(b) 度量学习前　(c) 度量学习后

图 4.7　应用度量学习的一个例子。

提高 TopicPanorama 中度量学习方法的速度，我们采用筛选法进行特征选择。筛选法的主要思想是计算每个特征与用户给定类别标签之间的相关程度，然后挑选出相关程度最高的那些特征作为结果。在 TopicPanorama 中，我们将用户给定的约束作为类别标签。具体来说，如果用户规定 v_i 和 v_j 要进行匹配，那么 (v_i, v_j) 将被标记为 1。如果用户规定 v_i 和 v_j 不要进行匹配，那么 (v_i, v_j) 将被标记为 0。在用户提供的类别标签不足时，我们会将不确定度最低（即节点替换代价最小）的匹配结果作为用户给定的约束，加入特征选择的过程中。为了计算特征和类别标签之间的相关程度，我们测试了三种被广泛应用的相关度指标：皮尔森相关度，互信息，以及 Relief[104]。相关的实验结果见第 4.6.3节。基于这个实验结果，我们采用互信息作为 TopicPanorama 中的特征选择方法。

4.4　全景图可视化

在这一节中，我们介绍 TopicPanorama 的可视化设计、布局算法以及交互。

4.4.1　可视化设计

我们基于第 4.1.1中对领域专家分析任务的调研结果进行了可视界面的设计。具体来说，可视化界面希望能够帮助用户轻松分析主题的层次结构、主题图的匹配结果以及主题匹配的不确定性。下面，我们对每个部分进行详细介绍。

4.4.1.1　用径向冰柱图展示主题层次结构

为了处理含有大量主题的文本集合，我们利用 Wang 等人提出的带约束的贝叶斯多分枝树算法[2] 对主题图中的主题进行聚类，生成主题树。主题树中，每一个叶子节点是一个主题，每一个非叶子节点是一个主题类。带约束的贝叶斯多分枝树算法采用贪心策略建立主题树，在每一次迭代中都寻找使得后验概率增加最大的两个子主题树进行合并。我们之所以采用这个算法，是因为它可以生成中间节点含有多个孩子的多分枝树结构，能更好地拟合文档中的真实主题分布。具体来说，我们首先对每一个主题图都生成一棵主题树，然后将其中一部分主题树作为约束，建立另外的主题树。这样迭代优化，直到算法收敛。我们采用径向冰柱树（图 4.9(b)）来显示主题的层次结构（**T4**）。这些径向冰柱树被布局在径向的最外侧，它所占的扇形角大小表示该文本集合中的主题个数。

4.4.1.2　用基于密度的图可视化展示匹配结果

现有研究表明，熟悉的可视化展现形式有利于减少用户的认知负担，并且有利于用户发挥自己的知识和经验来加速学习过程[105]。因此，我们设计的一个基本原则是在合适的情况下尽可能多地利用用户熟悉的可视化隐喻。在这个思想指导下，我们采用了叠加式的视觉比较方法。这类方法对于多张图的比较较为有效[44]。

基于上面原因，我们设计了一个基于密度的图可视化来展现用户选定层级上的匹配结果。这个图可视化包含两个部分，第一个部分是一个节点-链接图，第二个部分是一个密度图（图 4.9(e)）。当用户选定了感兴趣的层级后，我们从该层级含有的主题类中挑选出一些有代表性的主题，将这些主题和它们之间的关联展示在节点-链接图中。没有被选中的主题被分配到与它们最为相似的代表性主题上，然后以密度图的形式进行展现，给用户提供一个全局的上下文（**T1**, **T3**）。在节点-链接图中，不同文本集合的主题用不同的颜色标识。布局在中央位置的是多个文本源共有的主题，每个共有主题用一个饼图表示，饼图中的几种颜色代表这个主题被哪些文本集合共享（图 4.9，**T2**）。每个主题都被布局在离自己文档集合对应的径向冰柱树尽可能近的位置。共有的主题被多个径向冰柱树吸引，这些吸引力互相抵消，因此共有主题往往布局在视图中央（图 4.9）。在视图中，被相同文档集合包含的主题会布局在接近的位置。除此之外，同一个主题类中的主题也会被布局在接近的位置。我们的用户都比较喜欢这个既可以展示有代表性的细节，又可以展示全局上下文的混合的可视化设计。

图 4.8　展现不确定性的符号。

4.4.1.3　用符号表示不确定性

在看过我们第一个版本的系统后，专家们发现了一些不太正确的匹配结果。他们表示希望能够显示分析当前匹配结果的不确定性。这与之前的研究结果相符。之前的研究表明，更好地分析不确定性对于用户的数据分析非常重要[106,107]。为了帮助用户更好地进行分析，我们设计了一个如图 4.8所示的符号来表示匹配结果的不确定性。这个符号是受象形符号（Iconic Symbol）中“Filled Bar and Slider”的启发设计的。“Filled Bar and Slider”是众多表示属性不确定性的方法中效果较好的

一种[108]。在这个视觉隐喻中，我们用两个红色三角之间的扇形角度大小来表示不确定程度，扇形角越大代表匹配的不确定性越大。

4.4.2　布局算法

在这一节中，我们主要介绍主题树与基于密度的图可视化的布局算法。

4.4.2.1　主题树布局

给定 N 个文本集合，径向冰柱树的布局较为简单直接。我们将不同文本集合的独有主题和所有文本集合的共有主题布局在径向冰柱树的中央位置。其他少于 N 个文本集合共有的主题被布局在与各个文本集合的径向冰柱树尽可能近的位置。

4.4.2.2　基于密度的图可视化布局

我们将这个布局问题建模成带约束的弹簧模型。这个模型可以较好地平衡布局结果的可读性与稳定性。

可读性。 我们用两个衡量标准来计算匹配图布局结果的可读性。第一个衡量标准是 Kamada 等人[109] 提出的两个点之间实际距离与它们图上理论距离的差别 $E_k = \sum_{v_i \neq v_j} \left[(|p_i - p_j| - l_{ij})^2 / l_{ij}^2 \right]$。这里，$p_i$ 代表节点 v_i 在匹配图中所在的位置，l_{ij} 是 v_i 与 v_j 的图上理论距离。第二个衡量标准是基于格式塔理论中的接近律定义的，它是为了确保节点在匹配图中的位置和它在径向冰柱树上的位置比较接近。具体来说，它定义为 $E_h = \sum_{v_i} |p_i - \hat{p}_i|^2$。另外，为了满足上面提到的设计需求，我们定义了两个必须满足的约束：

- 文本集合约束 C_p：被同样文本集合共享的主题要布局在同一个区域，我们称这个区域为集合区域。
- 类别约束 C_l：在同一个集合区域中，属于同一个类别的主题要布局在同一个区域，我们称这个区域为类别区域。

稳定性。 为了确保视图在不同层级间进行切换时，用户能够更好地跟踪视图的变化，我们定义了 E_s 来衡量两个视图之间的稳定程度。根据 Misue 等人[110] 的研究成果，确保节点在水平方向和竖直方向的相对位置对于用户更好地跟踪视图变化非常重要。受此影响，我们在衡量稳定性时主要判断每两个节点之间在水平方向和竖直方向的相对位置是否有变化。如果两个节点在竖直方向（或者水平方向）的相对位置发生了变化，我们将稳定性损失定义为 v_i 和 v_j 之间在竖直方向（或者水

平方向）上距离的平方：

$$E_s = \sum_{(x_i - x_j)(x_i' - x_j') < 0} (x_i - x_j)^2 + \sum_{(y_i - y_j)(y_i' - y_j') < 0} (y_i - y_j)^2 \tag{4-11}$$

这里 x_i/y_i 是 v_i 的 x/y 坐标，x_i'/y_i' 是 v_i 之前布局中的 x/y 坐标。

带约束的弹簧模型。 我们利用带约束的弹簧模型统一表示上述衡量标准与约束：

$$min\ E = E_k + \lambda_h E_h + \lambda_s E_s,\ s.t.,\ C_p,\ C_l \tag{4-12}$$

这里，$\lambda_h > 0$，$\lambda_s > 0$ 是用来平衡 E_k、E_h 和 E_s 的参数。TopicPanorama 中，我们将 λ_h 设为 1，λ_s 设为 0.5。

公式 (4-12) 中的能量函数 E 可以利用三个弹簧力进行局部优化。第一个力 F_k 是 Kamada 等人[109] 提出的匹配图中两个主题之间的弹簧力，它可以用来最小化能量 E_k。第二个力 F_h 是匹配图中的主题和径向冰柱树上对应主题之间的弹簧力，它可以用来最小化能量 E_h。第三个弹簧力 F_s 是加在匹配图中相对位置发生了变化的主题之间的，被用来最小化能量 E_s。为了满足约束 C_p 和 C_l，我们用 Voronoi 剖分来确定集合区域与类别区域。我们之所以选择 Voronoi 剖分，是因为两个原因[111]：1) 它生成的布局区域的长宽比较优，趋近于 1；2) 它可以考虑集合区域或者类别区域的权重，给包含更多节点的区域分配更大的空间。为了对带约束的弹簧模型进行求解，我们将 Voronoi 剖分和力引导图布局方法相结合，计算匹配图上节点的坐标。具体来说，布局算法主要分为以下四步。

第一步主要计算各个集合区域。具体来说，我们首先将属于每个集合区域的节点合并成一个节点，连接不同集合区域的边合并成一条边，形成一幅元图。然后，我们把元图中的节点和相应的径向冰柱树进行连接，并利用力引导模型对元图中的节点进行布局（图 4.9(b)）。这些节点的位置就是相应集合区域的中心。我们在这些中心的基础上进行 Voronoi 剖分，计算出各个集合区域。算法第二步主要计算各个类别区域。具体来说，我们在每个集合区域中放置用户选定层级上的主题类节点。利用类似第一步中的力引导算法，我们计算出了这些主题类节点的位置，然后基于这些节点的位置计算 Voronoi 剖分得到类别区域（图 4.9(c)）。算法的第三步是计算代表性节点的位置。对于每一个类别区域，我们挑选出一些具有代表性的节点。我们用到的是 TIARA[17] 中提出的主题排序算法。该算法主要考虑了主题的覆盖率（Coverage）、方差（Variance）以及独特性（Distinctiveness）进行主题的挑选。这个算法挑选出来的主题能够较好地覆盖所在类别的内容（覆盖率），

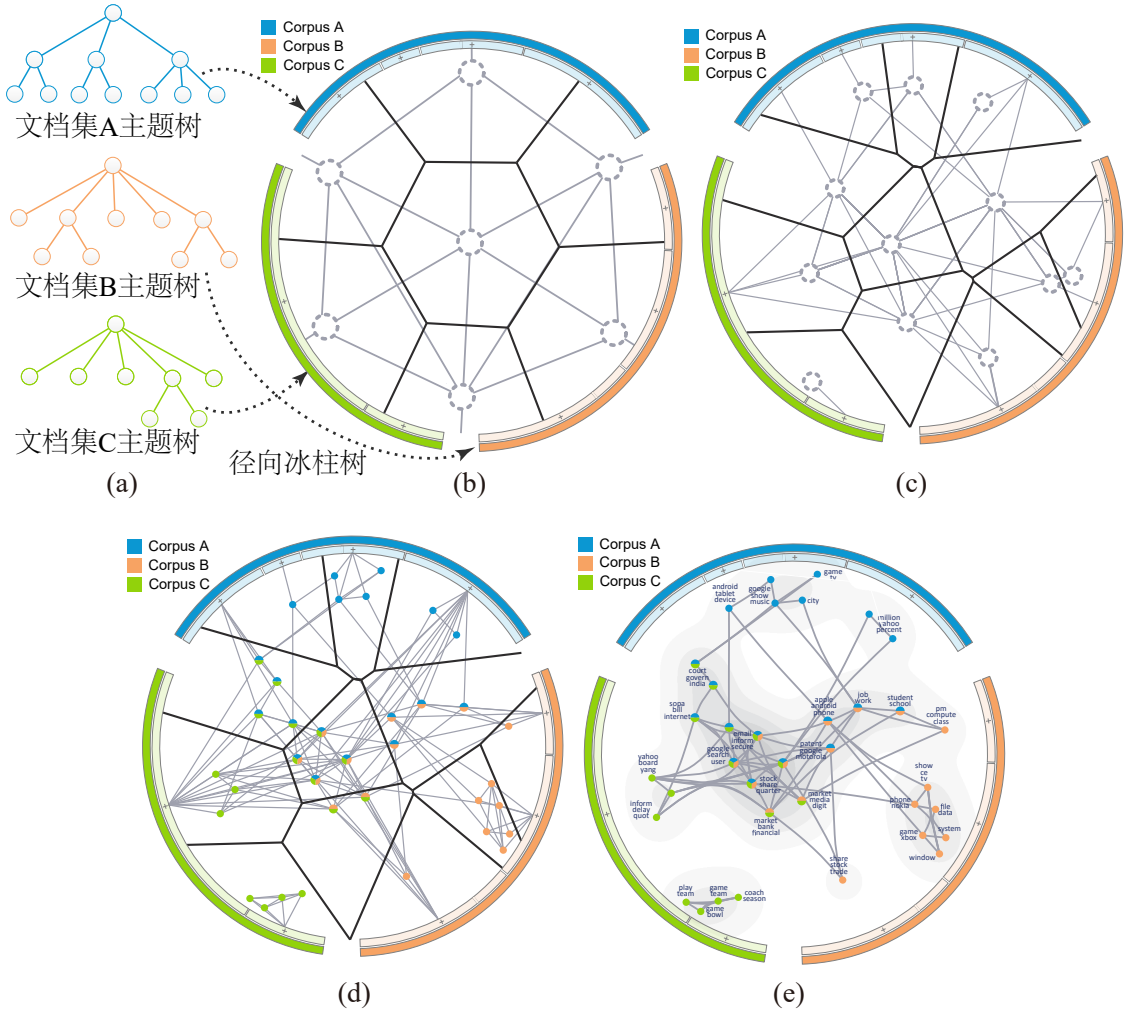


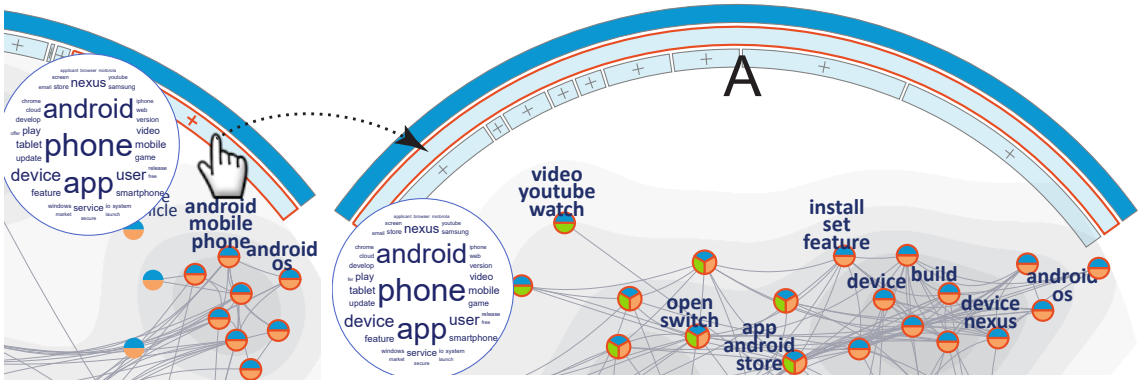
图 4.9　布局算法的主要思想：(a) 不同文本集合的主题树；(b) 对不同文本集合的共有部分和独有部分进行布局，并且计算 Voronoi 剖分；(c) 对用户选定层级的主题类节点进行布局，保证它们布局在对应 Voronoi 单元的内部，并且根据新的布局结果重新计算 Voronoi 剖分；(d) 对选择出的代表性主题进行布局；(e) 最终的布局结果。

不会在所有类别中都频繁出现（方差）以及与其他挑选出来的主题差别较大（独特性）。我们利用如第二步中的力引导算法将这些被挑选出来的主题布局在相应类别区域中（图 4.9(d)）。算法的第四步中，我们将非代表性主题对应到最相似的代表性的主题，然后利用核密度估计法（Kernel Density Estimation, KDE）[112] 对全局上下文进行可视化（图 4.9(e)）。

4.4.3　交互

为了帮助用户更好地分析主题全景图，我们给用户提供了下面的交互。

自由探索主题树的不同层级（T4）。用户可以通过点击径向冰柱树上的相应节点来放大到感兴趣的主题类。这个交互具体是如下进行的。首先，一个用户可以将鼠



(a)　　(b)

图 4.10　探索主题树的不同层次。

标悬停在径向冰柱树上的节点。这个节点附近会出现一个词云，显示出这个节点对应的主题类的内容（图 4.10(a)）。这个主题类在匹配图上对应的节点会被高亮。基于这些提示，一个用户可以决定他/她是否希望放大探索相应的主题类。如果用户对这个主题类感兴趣，他/她可以点击这个节点放大相应主题。如图 4.10(b) 所示，放大后，这个类别中的更多的主题会被显示出来。另外，径向冰柱树上被点击的节点的孩子也被显示出来（**A**），方便用户进行进一步探索。我们利用分阶段的动画[113] 让用户在放大/缩小过程中更容易跟踪视图的变化。

分析感兴趣的主题和它们之间的关联（T3）。为了保证用户更好地分析他们感兴趣的主题，我们提取了主题对应的关键词和代表性的文档。用户可以通过点击匹配图上的节点来分析它对应的关键词（图 4.1(e)）和文档（图 4.1(f)），或者利用拉索选择一组主题来分析这组主题的分布。在用户发现一个感兴趣的主题以后，他/她可以高亮与这个主题相关的其他主题，从而找到更多感兴趣的主题。如果用户将鼠标悬停在匹配图中的主题节点上，这个主题在径向冰柱树中的位置会被高亮。我们还允许用户利用搜索操作来找到感兴趣的主题。

分析匹配主题的领先-滞后关系（T5）。为了帮助用户更好地分析匹配主题在不同文本集合随时间变化的规律，TopicPanorama 对用户选定主题的领先-滞后关系进行了可视化。领先-滞后关系显示了在特定共有主题上，哪个文本集合（领先）被其他文本集合（滞后）追随。我们利用 Liu 等人[3] 提出的算法来计算领先-滞后关系随时间的变化。给定两个文本集合和特定时刻 t，当且仅当文本集合 A 在 t 时刻的内容与其他文本集合在 t 时候之后（而不是之前）的内容相似时，该算法认为文本集合 A 在 t 时刻领先。Liu 等人的算法[3] 只能展示两个文本集合领先-滞后关系随时间的变化。我们改进了他们的算法，设计了一个基于线图的可视化来展示多个文本集合之间领先-滞后关系随时间的变化（图 4.11））。在这个视图中，每条线代表一个文本集合，x 轴代表时间，y 轴代表文本集合的领先程度，在上方的线对

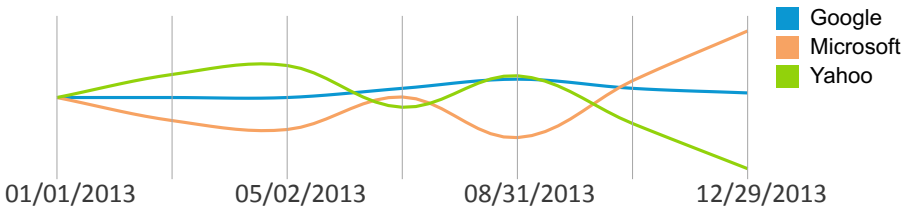


图 4.11　随时间变化的领先-滞后关系视图。

应的文本集合领先于下方的线对应的文本集合。线相交表示文本集合的领先-滞后关系发生了变化。

交互式图匹配结果修改（T6）。受 Endert 等人提出的语义交互（Semantic Interactions）的启发，我们允许用户直接在视图上进行操作，对匹配结果进行修改。我们支持下面三种操作：匹配若干主题，分离匹配的主题，以及确认匹配结果正确。前两个操作用来对错误的匹配结果进行修改，最后一个操作用来降低正确匹配结果的不确定程度。在匹配结果修改算法中，确认匹配这个操作与匹配若干主题这个操作被同等对待。具体来说，在估计用户心中理想距离 d_t 时，都是用到一个小于 1，大于 0 的 α 值。在用户进行修改以后，全景图会被相应地更新（图 4.13）。用户在修改匹配结果的过程中，修改操作会被记录在右侧的匹配修改历史面板中（图 4.1(c)）。这个面板使得用户可以方便地对修改操作进行撤销和重做。另外，这个面板还支持用户检查特定修改操作影响了哪些主题。用户可以通过点击右上角的眼睛形状的图标（**G**）寻找不确定度较大的、最可能匹配错误的主题。我们利用一个 Scented 部件[114]（**H**）来帮助用户进行不确定匹配的筛选。

4.5　实现细节

在这一节中，我们主要介绍一些系统实现细节。

4.5.1　主题图的生成

我们利用两个方法建立主题图。第一个方法是并行相关图模型 CTM。我们利用这个方法建立如新闻的长文本的主题图。第二个方法是将 CTM 与后处理方法结合，建立短文本的主题图。后处理方法是利用短文本之间的链接（例如推特之间的相互引用关系）和短文本主题之间的相似度来提取主题之间的关联程度。下面，我们对这两种方法进行具体的介绍。

并行相关图模型。 我们利用一个近期提出的并行相关图模型[97] 来建立主题图。该算法提出了一种快速的吉布斯采样方法，可以快速从百万篇文档中提取出来上千主题。这个算法的主要思想是通过引入 Polya-Gamma 辅助变量，将非共轭

关系转化为条件共轭关系，从而使得并行的吉布斯采样变为可能。

并行相关图模型与后处理结合。 尽管并行相关图模型在大多数文本集合上效果较好，但是在短文本（如推特）上的结果还有所欠缺。为了解决这个问题，我们利用推特中包含的丰富元数据，即推特的引用关系、共用标签关系等，与主题相似程度结合来提取主题之类的关联，提高主题图提取的准确性。

4.5.2　主题树的构建

为了方便用户对含有大量主题的主题图进行分析，我们利用贝叶斯多分枝树[5,86] 来建立主题图的层次结构。为了保证不同文本集合的主题图层次结构较为相似，我们利用了带约束的贝叶斯多分枝树[2]。这个方法可以以其他主题图的层次结构为约束，迭代地对当前主题图的层次结构进行优化，使得在不牺牲它层次结构的准确性的同时，尽量跟其他主题图的层次结构保持一致。

4.6　数值实验

我们设计了三个数值实验，验证我们算法的有效性和效率。首先，我们说明用 CTM 为多个文本集合提取的统一的主题图并不能对每个文本集合都进行很好的拟合。然后，我们说明提出的图匹配算法与其他基准算法相比，在精确度（Precision）、召回率（Recall）以及一致性方面都更好。最后，我们验证提出的交互式图匹配修改算法的有效性与效率。

我们在实验中用到了下面的几个数据集：

- **数据集** A 是从 Boardreader[115] 中收集的。数据集中包含三个文本集合：一个新闻集合、一个博客集合和一个论坛集合。时间区间是从 2008 年 7 月到 2009 年 4 月。
- **数据集** B 是从 BingNews[85] 中收集的。它包含 2013 年 9 月至 12 月期间与三家 IT 公司（百度、阿里巴巴、腾讯）相关的中文新闻。
- **数据集** C 和**数据集** D 包含与 Google、Microsoft 和 Yahoo 三家 IT 公司相关的从 2013 年 1 月至 2013 年 12 月的新闻。数据集 C 是从数据集 D 中采样得到的。采样的目的是为了降低专家标注匹配结果的代价。
- **数据集** E 和**数据集** F 是用“Ebola”这个关键词从新闻和推特中收集的（2014 年 7 月 27 日至 2015 年 2 月 21 日）。数据集 E 是从数据集 F 中采样得到的。采样的目的是为了降低专家标注匹配结果的代价。

我们邀请两个文本挖掘专业的在读博士生对匹配结果进行了标注。他们之间有 83.8% 的标注结果是一致的。表 4.1对用到的数据集进行了总结。所有的实验都

在一个 CPU 为 2.4GHz 的 Intel Xeon E5620 、内存为 12GB 的工作站上进行的。

表 4.1　实验中用到的 6 个数据集的统计结果。$|\mathcal{D}|$：文本集合中文本个数；$|\mathcal{V}|$ 和 $|\mathcal{E}|$：主题图中的点的个数和边的个数。

(a) 数据集 A

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
新闻	26,538	60	68
博客	13,424	50	51
论坛	15,272	59	86

(b) 数据集 B

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
百度	16,723	100	345
阿里巴巴	12,925	100	336
腾讯	39,074	100	363

(c) 数据集 C

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
Google	54,338	93	152
Microsoft	37,001	115	230
Yahoo	1,701	112	176

(d) 数据集 D

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
Google	147,887	260	713
Microsoft	100,134	314	1285
Yahoo	6,280	246	872

(e) 数据集 E

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
新闻	100,450	108	161
推特	6,381,868	130	222

(f) 数据集 F

	$\|\mathcal{D}\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
新闻	207,406	324	817
推特	15,565,532	390	833

4.6.1　单一 CTM 模型的不足

4.6.1.1　实验设置

在这个实验中，我们用到了数据集 A、数据集 B 以及数据集 D。对于每一个数据集，我们都用 CTM 学习了四张主题图。每张主题图是 100 个主题。这四张主题图中，有三张是针对每个集合中的文档分别学习出来的（Sep.），有一张是针对所有文档学习出来的（Joint）。

4.6.1.2　结果

表 4.2显示了每张主题图对不同文档集的拟合程度。这里，我们利用 Perplexity 值来衡量主题图对文档集的拟合程度。Perplexity 值越高，模型困惑度越大，对数据集的拟合程度越小。实验结果证明单独学习的主题图比针对所有文档训练的主题图的拟合度更高。这个结果也说明，每个文档集合的内容有差异。因此，我们需要用图匹配法把每个文档集合的主题图拼接在一起，得到更全面的主题全景图。

表 4.2　针对每个集合的文档分别训练的主题图（Sep.）以及针对所有文档训练的主题图（Joint）对各文档集合的拟合程度。拟合程度由 Perplexity 值衡量。Perplexity 值越高，对数据集的拟合程度越小。可以看出，分别训练的主题图对数据的拟合程度比利用所有文档集合训练的主题图（单一 CTM 模型）对数据拟合程度更好。

	数据集 A			数据集 B			数据集 D		
	新闻	博客	论坛	百度	阿里巴巴	腾讯	Google	Microsoft	Yahoo
Sep.	**2898**	**3792**	**2333**	**2017**	**2022**	**2031**	**2604**	**2203**	**1822**
Joint	3037	4058	2640	2055	2183	2093	2872	2444	2202

4.6.2　图匹配算法实验

4.6.2.1　实验设置

在这个实验中，我们在精确度、召回率与一致性等方面对比我们的算法和第 4.2.1节中的两个基准算法。这个实验中，我们用到了两个由专家标注的数据集：数据集 A 以及数据集 C。

表 4.3　我们的算法与两个基准算法的精确度、召回率、不一致匹配个数和运行时间（秒）对比。

	数据集 A				数据集 C			
	精确度	召回率	不一致	时间	精确度	召回率	不一致	时间
我们的算法	**0.81**	**0.79**	**0**	1.3	**0.79**	**0.92**	**0**	8.8
基准算法一	0.79	0.77	4	1.2	0.69	0.85	10	8.5
基准算法二	0.77	0.67	**0**	**0.8**	0.69	0.76	**0**	**5.7**

4.6.2.2　结果

如表 4.3所示，我们的方法在精确度、召回率以及不一致匹配个数方面都优于基准算法。基准算法一的精确度和召回率和我们的方法相当，但是它的结果中含有一些不一致的匹配。基准算法二可以生成一致的匹配结果，但是它的精确度和召回率是最低的。另外，我们算法的运行时间和两个基准算法基本相当。

4.6.3　交互式图匹配结果修改实验

4.6.3.1　实验设置

在这个实验中，我们用到了三个专家标注的数据集：数据集 A、数据集 C 以及数据集 E。对于用到了特征选择的方法，我们利用网格搜索（100, 110, ..., 200）

表 4.4　对比五种匹配结果修改算法的修改步数（$|\mathcal{U}|_{avg}$，$|\mathcal{U}|_{min}$ 和 $|\mathcal{U}|_{max}$）和运行时间。这里 NoML 和 ML 分别指没用度量学习和用了度量学习的方法。NoFS 指没有用特征选择的方法，PC、MI 和 Relief 分别指用到皮尔森相关性、互信息以及 Relief 作为相关度测量的特征选择方法。

(a) 数据集 A

	$\|\mathcal{U}\|_{avg}$	$[\|\mathcal{U}\|_{min}, \|\mathcal{U}\|_{max}]$	时间
NoML	17	[17, 17]	0.0046
ML-NoFS	15.75	[15, 17]	2.9602
ML-PC	15.35	[15, 16]	0.0755
ML-MI	13.35	[11, 17]	0.085
ML-Relief	15.05	[13, 17]	0.3201

(b) 数据集 C

	$\|\mathcal{U}\|_{avg}$	$[\|\mathcal{U}\|_{min},\|\mathcal{U}\|_{max}]$	时间
NoML	35	[34, 36]	0.0085
ML-NoFS	23.6	[21, 26]	0.8382
ML-PC	23.2	[21, 25]	0.0366
ML-MI	23.1	[21, 26]	0.1062
ML-Relief	24.45	[23, 26]	0.0981

(c) 数据集 E

	$\|\mathcal{U}\|_{avg}$	$[\|\mathcal{U}\|_{min},\|\mathcal{U}\|_{max}]$	时间
NoML	28.4	[28, 29]	0.0018
ML-NoFS	23.15	[20, 28]	342
ML-PC	22.8	[21, 24]	0.1145
ML-MI	19.65	[19, 20]	0.2113
ML-Relief	19.75	[18, 24]	0.3154

来确定挑选出的特征个数。

4.6.3.2　评价标准

我们用算法的平均响应时间作为算法效率的衡量标准。为了衡量算法的有效性，我们用到的是算法将所有错误匹配修正所需要的约束个数（$|\mathcal{U}|$）。具体来说，我们首先将匹配算法的结果和用户标注过的正确匹配结果进行对比，找出算法产生的错误匹配。然后，我们从错误的匹配中随机挑选出一个，将其对应的正确匹配结果当作约束，输入到匹配修改算法。匹配修改算法利用这个约束更新匹配结果。我们不断重复上述操作，直到算法输出的匹配结果完全正确为止，并记录下来总共输入修改算法的约束个数。约束个数越少，修改算法越高效。为了消除约束输入

顺序造成的随机误差，我们进行了 20 次随机实验，并将这些实验的约束个数的平均值（$|\mathcal{U}|_{avg}$）、最小值（$|\mathcal{U}|_{min}$）以及最大值（$|\mathcal{U}|_{max}$）记录下来，作为最终实验结果。

4.6.3.3　结果

表 4.4显示了五种匹配结果修改算法的有效性和效率。我们的分析结果如下。

用度量学习的方法与没用度量学习的方法的对比。 表4.4显示，用到了度量学习的方法比没有用到度量学习的方法（NoML）效率明显要高。这说明，用到了度量学习的方法可以从输入算法的约束中更好地学习节点替代操作的代价，从而生成更准确的匹配结果。但是，用到度量学习的方法比没有用到度量学习的方法要慢一些。如 ML-NoFS 方法在数据集 E 上用时 342 秒，远远超过了实时修改所能接受的算法响应时间。这说明，度量学习方法的一个主要问题是时间复杂度较高。

用特征选择的方法和没用特征选择的方法的对比。 从表4.4可以看出，用到了特征选择的方法和没有用到特征选择的方法的有效性一样好。在很多情况下，用到了特征选择的方法的有效性甚至更好一些。这说明我们用到的特征选择方法能够较好地去除掉特征中的噪音。另外，用到特征选择的方法比没有用到特征选择的方法再运行效率上至少高八倍。其中，最慢的响应时间是 0.3201 秒（数据集 A，ML-Relief）。这种响应时间支持用户对图匹配结果进行实时修改。综合来说，用到特征选择的方法可以在不损失算法有效性的同时大大提高算法效率。

不同特征选择方法的对比。 在所有的基于特征选择的方法中，ML-MI 略优于其他算法。ML-MI 比 ML-PC 的结果好是因为互信息是一种更加全面的相关性衡量标准。皮尔森相关性主要用于测量连续的实数变量之间的线性相关性，而互信息也可以测量离散变量之间的相关性以及更高维度的相关性。ML-MI 比 ML-Relief 的结果更好，是因为在我们的应用中，ML-Relief 所需要的训练样本不足，因此结果不够准确。ML-Relief 与 ML-MI 的主要区别在于它还可以考虑不同特征之间复杂的相关关系，而 ML-MI 假设特征之间是独立、无关的。因此，ML-Relief 所需要的训练样本数量更多。在我们的应用中，用于提取特征的类别标签即用户提供的约束，数量较少，不足以让 ML-Relief 的方法学习出来很好的结果。

4.7　案例分析

我们与领域专家进行密切合作，进行了下面的案例分析。

4.7.1　IT 公司

这个案例分析的主要目的是诠释 TopicPanorama 如何帮助用户完成分析任务，并且指出系统中哪些功能对完成相应的任务起到了关键作用。这个案例分析中主要用到了表 4.1(d) 中所示的数据集 D。数据集 D 中，包含与三家 IT 公司（Google、Microsoft 以及 Yahoo）相关的新闻。一个担任了 10 年公共关系经理的专家（P1）参与到了案例分析中。她用了两个小时的时间来进行案例分析，期间我们对她进行了为数不多的引导。

概览（任务 T1 与 T2）。我们首先给专家看的是这三家公司的主题全景图的概览（图 4.1(a)）。从这个概览中，专家立即辨别出了不同公司的共有主题与独有主题。例如，与搜索和市场相关的主题是三家公司共有的（**A**）。大多数与手机相关的主题是 Google 和 Microsoft 共有的，小部分是三家公司共有的（**B**）。与政府相关的主题部分由三家公司共有，部分由 Google 和 Yahoo 共有（**C**）。与汽车相关的主题主要在 Google 相关新闻中提到（**D**）。与 Kinect 相关的主题主要在 Microsoft 相关新闻中提到（**E**）。Yahoo 相关新闻中有一部分独有主题是与它的 CEO，Marissa Mayer 相关的（**F**）。

从多层级探索与政府相关的主题，并分析这些主题随时间变化的规律（任务 T4 与 T5）。专家对为什么有这么多与政府相关的主题很感兴趣。因此，她通过点击径向冰柱树上的节点不断放大相关主题，直到进入到最底层。这里面，大部分主题是与 NSA 窃听丑闻相关的。她发现这些主题中，有一部分是与三家公司都相关的。在她的经验中，这些主题往往更加重要，因此她仔细研究了这些主题。如图 4.12(a) 所示，这些主题可以分为两组。第一组包含一个主题（**G**），这个主题讨论的是丑闻首次被披露时的情况。第二组包含三个主题（**H**、**I** 和 **J**）。这三个主题是这三家公司针对丑闻的反应，从它们是共有主题来看，这三家公司对丑闻的反应非常类

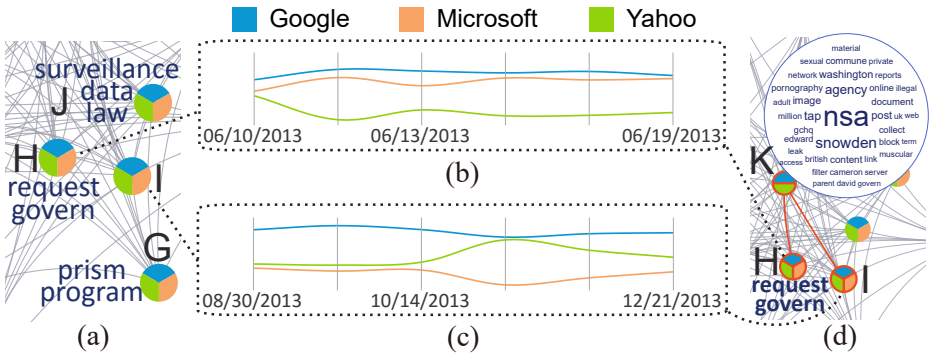


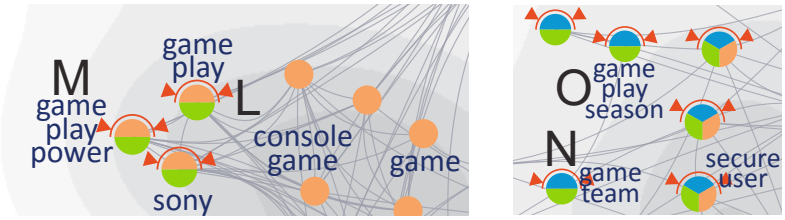
图 4.12　NSA 窃听丑闻相关主题中被 Google、Microsoft 以及 Yahoo 共享的主题：(a) 相关主题；(b) 主题 **H** 上，不同文档集合的领先-滞后关系；(c) 主题 **I** 上，不同文档集合的领先-滞后关系；(d) 一个与 **H** 和 **I** 都相关的主题。

似。首先，它们都否认与政府进行过合作，并通过披露政府索要数据的数目来博取公众的信任（**H**）。图 4.12(b) 显示出了这个主题中，不同公司的领先-滞后关系。可以看出，Google 和 Microsoft 比 Yahoo 更早地向大众披露了政府索要数据的信息。接着，这三家公司都对自己的数据中心之间的通信渠道进行加密（**I**）。如图 4.12(c) 所示，Google 在这项工作中处于领先地位，Yahoo 在它的后面，而 Microsoft 是这三家公司中最后一个进行数据加密的。专家最开始认为只有 Google 和 Yahoo 对数据进行了加密，我们的系统帮助她发现 Microsoft 也对数据进行了加密。她说很高兴系统纠正了她原先的错误认识。最后，这三家公司和其他 IT 公司要求美国政府修改监管法（**J**）。

检查主题之间的关联（任务 T3）。在上面的分析中，专家发现了一个有趣的现象。在披露政府索要数据的信息时，Yahoo 是三家公司中最落后的。而在对数据进行加密时，Yahoo 却比 Microsoft 更为活跃了。专家对于 Yahoo 变活跃的原因很感兴趣，因此她高亮了与 **H** 和 **I** 都相关的主题。通过分析这些相关主题，她发现了主题 **K** 解释了 Yahoo 变活跃的原因。主题 **K** 是关于 Google 和 Yahoo 的数据传输通道被 NSA 入侵的（“NSA statement on Washington Post report on infiltration of Google, Yahoo data center links”）。因为 Yahoo 的数据传输通道被入侵了，为了让用户安心，Yahoo 积极地对数据进行了加密。因为 Microsoft 的数据传输通道没有被入侵，因此它不如 Yahoo 那样积极。

定制主题全景图（任务 T6）。专家对于游戏相关主题比较感兴趣，因此她在搜索框中输入了“game”进行查询。一部分查询结果如图 4.13(a) 所示。为了观察这里面是否有错误的匹配，她将匹配的不确定性进行了显示。通过观察不确定性较大的结果，她找到了两个错误的匹配，**L** 和 **M**。这两个主题匹配错误的原因是由“game”的多义性造成的。在 Microsoft 的相关主题中，“game”是指视频游戏，而在 Yahoo 的相关主题中，“game”是指体育赛事。通过修改 **M**，她发现 **M** 变成了 **O**，而 **L** 变成了 **N**（图 4.13(b)）。**O** 和 **N** 正确地将 Google 中与体育赛事相关的主题和 Yahoo 中与体育赛事相关的主题进行了匹配。通过用度量学习的方法，这两个错误的匹配只需要用户进行一次操作就可以全部纠正，而没有用度量学习的基于规则的方法[99] 则需要用户进行两次操作。

比较不同文本源的主题分布规律（任务 T1）。除了新闻中的主题以外，专家还很希望了解推特中与三家公司相关的主题。因此，我们为她提供了这三家公司在推特上的主题全景图概览（图 4.14(a)）。通过观察这幅图，她发现推特中主题的相关性不如新闻中主题相关性那么密切。另外，推特中共有主题的数目也更少（**A**、**B** 和 **C**）。例如，在与 Nexus 相关的主题中，大部分推特主题都是谷歌独有的（图 4.14(b)），而

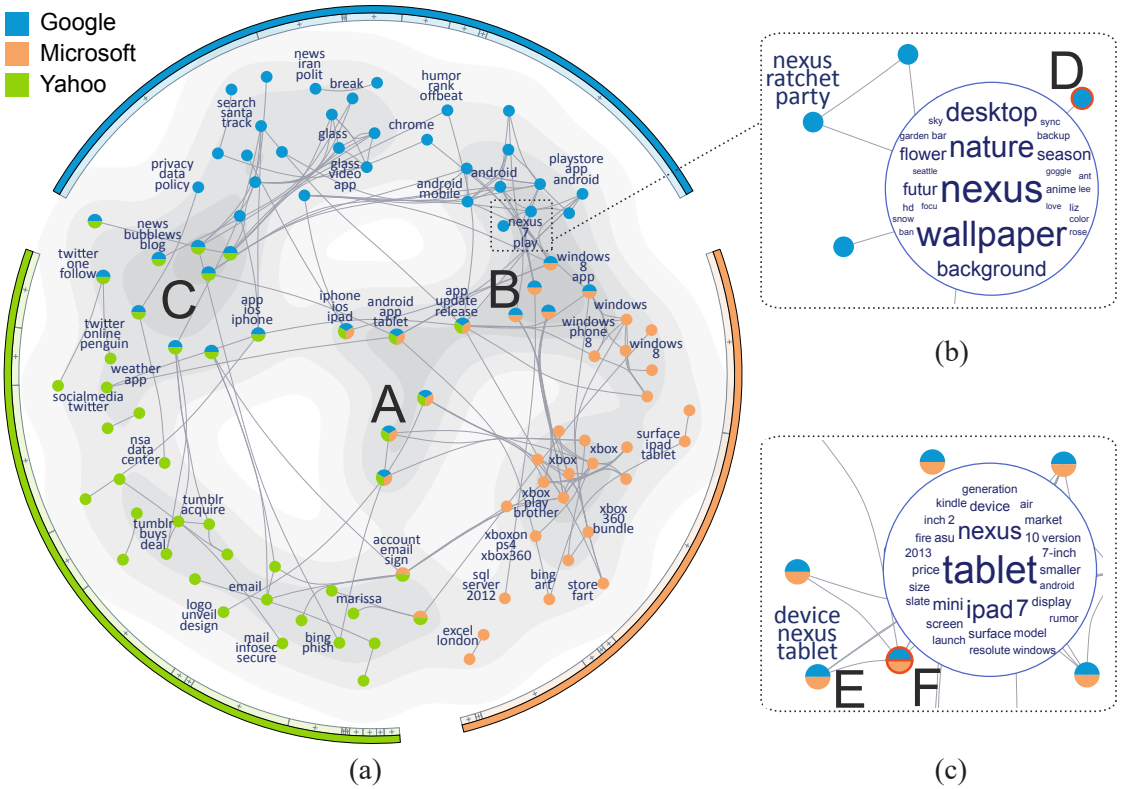


(a)　(b)

图 4.13　对匹配结果的交互式修改。

大部分新闻主题是 Google 和 Microsoft 共有的（图 4.14(c)）。通过对主题内容进行分析，我们发现这是因为 Nexus 相关的推特往往主题比较单一，都是跟 Nexus 的具体功能相关的（**D**）。而 Nexus 相关的新闻主要讨论的是 Nexus 的发布（**E**，“Google to launch new Nexus 7 tablet in July for $229: Report’’）以及与其他公司同类产品的比较（**F**，“New Nexus 7 vs iPad Mini. Screen Resolution Price and Specs’’）。因为 Microsoft 也有手机产品，因此 Google 和 Microsoft 共享这个主题。

对比不同文本源的共有主题与独有主题（任务 T2）。为了更好地对比新闻和推特中的主题，我们将新闻中的全景图和推特中的全景图进行了匹配。专家发现，推特中的主题比新闻中的主题数量更多（图 4.15(a)）。通过简单的探索，她发现 Tumblr



(a)　(b)　(c)

图 4.14　与 Google、Microsoft、Yahoo 相关的推特数据分析。(a) 推特数据集上的全景图概览；(b) 与 Nexus 相关的推特主题；(c) 与 Nexus 相关的新闻主题。

相关的主题是一个典型的推特主题比新闻主题多的例子，因此她仔细研究了这些主题。她发现与 Tumblr 相关的主题中，只有一个是新闻和推特共有的，其他主题都是推特独有的。这个共有主题是与 Tumblr 被 Yahoo 收购相关的（**A**）。这些独有主题主要是在发表自己的看法（**B**，“this whole yahoo and Tumblr relationship is painful. I don’t want it”）以及提供详细信息与建议（**C**，“Three Ways Yahoo Can Avoid Screwing Up Tumblr”）。研究这些推特主题以后，专家评论到，“看到这么多关于 Tumblr 收购的意见和看法非常有用，这可以帮助公司更好地采取措施。”

4.7.2　埃博拉

这个案例分析是跟教授 P2 合作完成的。她对于公共危机（例如埃博拉）中新闻媒体对民众的影响很感兴趣。表 4.1(e) 显示了在这个案例分析中用到的数据集。**概览（任务 T1 和 T2）**。首先，我们给教授提供了埃博拉相关文本集中主题全景图概览（图 4.16）。这里，下标 **n**、**t** 和 **c** 分别指的是新闻独有主题、推特独有主题以及新闻和推特共有主题。通过分析图中的关键词，她发现这些主题可以被分为四类：

- 埃博拉在西非的爆发（$\mathbf{A_n}$、$\mathbf{A_c}$ 和 $\mathbf{A_t}$）。相关主题提到了埃博拉疫情最严重的一些西非国家，包括利比亚（Liberia）、尼日利亚（Nigeria）以及塞拉利昂

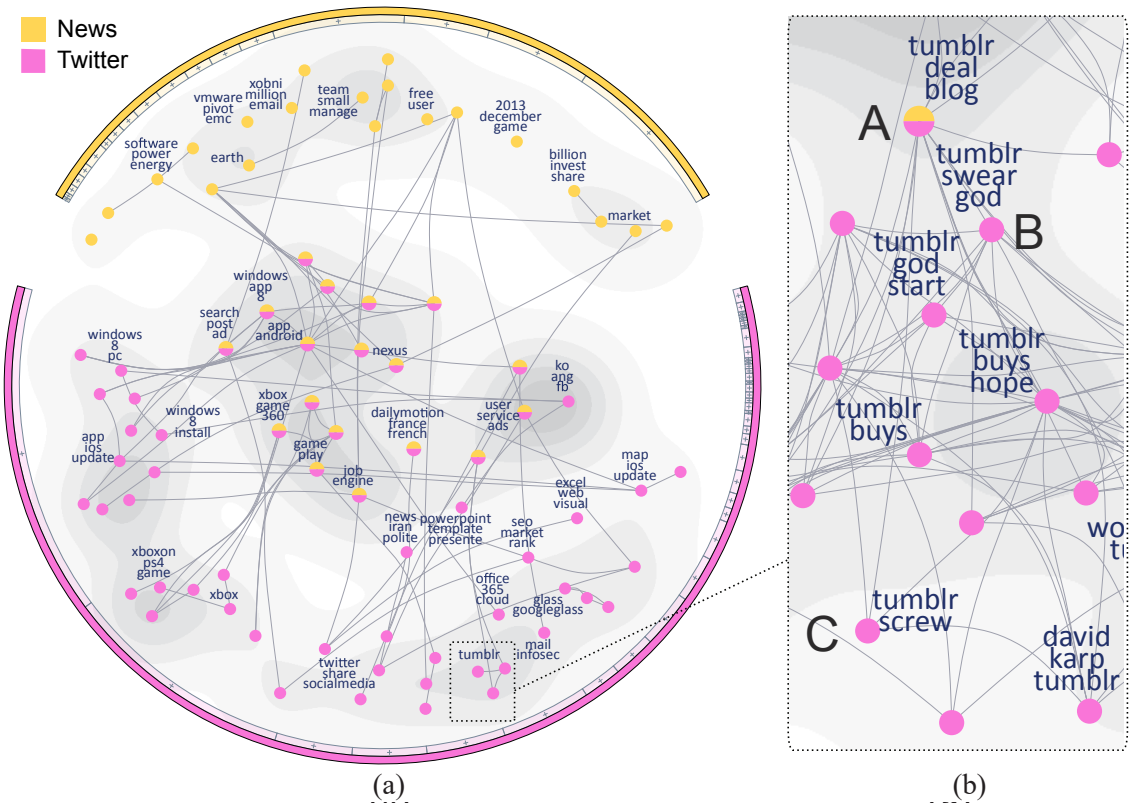


图 4.15　将新闻与推特进行匹配：(a) 概览；(b) 对比新闻中与 Tumblr 相关的主题和推特中与 Tumblr 相关的主题。

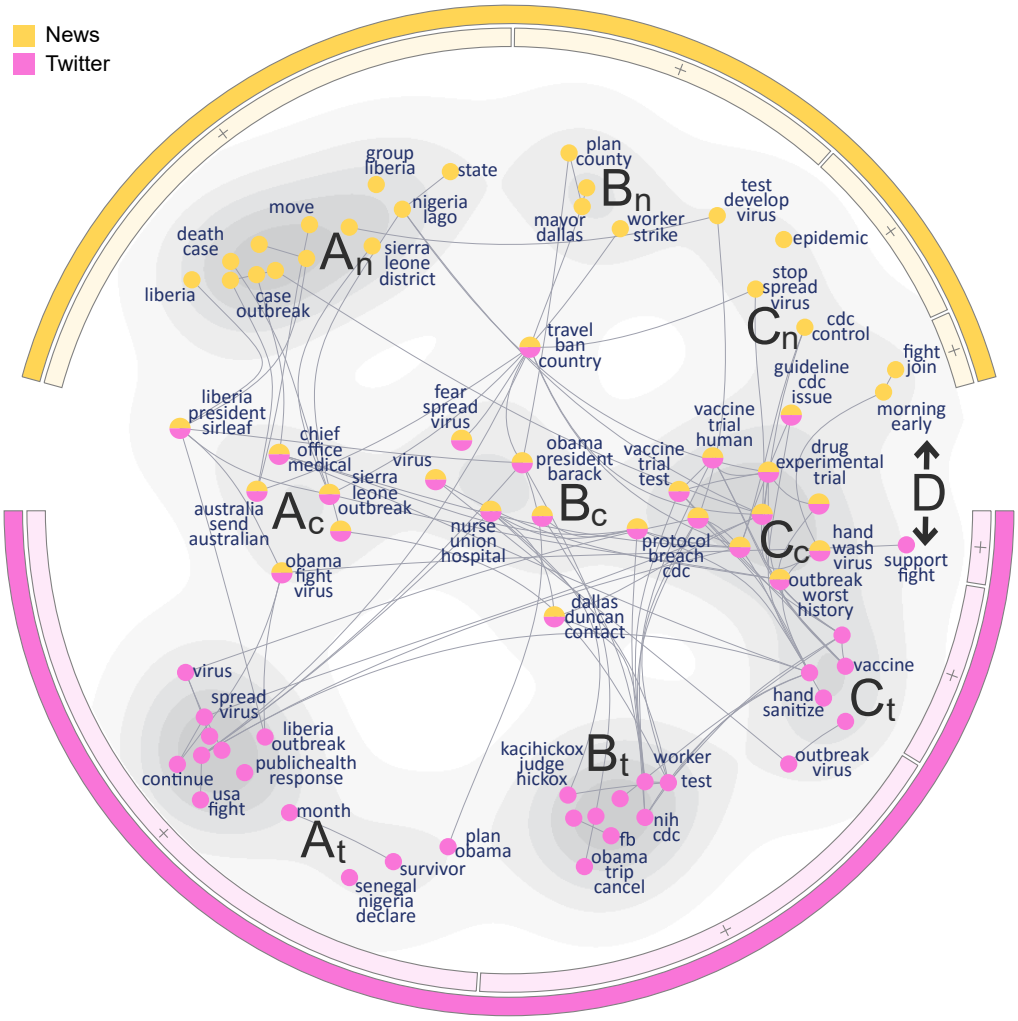


图 4.16　埃博拉相关新闻与推特主题全景图概览。这里，下标 **n**、**t** 和 **c** 分别代表新闻独有主题、推特独有主题以及新闻和推特共有主题。

（Sierra Leone）。新闻独有主题（$\mathbf{A_n}$）更多地提到了死亡病例（关键词“death”和“case”）。推特独有主题（$\mathbf{A_t}$）更多地提到美国是如何帮助西非国家与埃博拉进行抗争的（关键词“usa”、“fight”和“obama”）。

- 美国的埃博拉病人和疑似病例（$\mathbf{B_n}$、$\mathbf{B_c}$ 和 $\mathbf{B_t}$）。第二类中有关键词“dallas”、“nih”和“cdc”，这些都是美国的地点或者组织。新闻独有主题（$\mathbf{B_n}$）主要侧重于政府的计划和采取的措施（关键词“plan”、“county”和“mayor”）。推特独有主题 $\mathbf{B_t}$ 主要是对具体事件的讨论，例如奥巴马取消原来的行程，改为与内阁成员讨论埃博拉疫情（关键词“obama”、“trip”和“cancel”）。
- 与埃博拉相关的知识（$\mathbf{C_n}$、$\mathbf{C_c}$ 和 $\mathbf{C_t}$）。第三类中的主题主要是与埃博拉相关的知识。教授意识到这一类是共有主题最多的一类。这一定程度上说明了民众能较好地接受新闻中关于埃博拉的知识。
- 加入与埃博拉的斗争（**D**）。这一类中含有关键词“fight”和“join”，主要是

号召大家积极捐款，为政府和民间组织与埃博拉的斗争贡献自己的力量。

放大研究与埃博拉知识相关的主题（任务 T4）。教授想了解新闻中关于埃博拉的知识如何影响了公众的态度和行为。因此，她放大研究了第三类主题：与埃博拉相关的知识。通过不断点击径向冰柱树上与该主题相关的节点，她放大到了一个与埃博拉病毒相关的主题类（图 4.17(a)）。

为了了解新闻媒体如何对大众产生影响，她进一步研究了共有主题。大部分共有主题都是关于埃博拉病毒的传播方式的。例如，主题 **E** 中提到“How is Ebola spread? CDC expert explains ‘direct contact’ with bodily fluid”。通过探索，教授发现有一个主题反映了新闻媒体对大众的态度与行为的影响（**F**）。这个主题含有关键词“hand”与“wash”，是关于洗手的。这个主题中的新闻主要强调洗手对于预防埃博拉的重要性（“Preventing cholera, Ebola: Hand washing should be our top priority”）。通过检查相关推特以及查看领先-滞后关系（图 4.17(d)），我们可以看到新闻中的知识确实影响到了民众的态度和行为。

两个与主题 **F** 关联的推特独有的主题吸引了教授的注意。通过对它们进行检查，教授发现新闻中与洗手相关的主题造成了一定程度的恐慌。民众产生了一些不理智的行为，例如有人发推特称，“I be pouring like a gallon of hand sanitizer on my hands after I shake up with Africans now BC of that Ebola shit.”教授评论到：“在危机沟通中，政府与主流媒体首先需要告诉民众他们需要注意与警惕，然后还需要让民众保持冷静。可以看出新闻媒体在第一步上很成功，但是在第二步上做得还不到位。”教授对于媒体如何更好地安抚民众提出了两条建议。

- **给出关于如何洗手的具体、可行的指导。** 除了号召民众进行洗手，新闻媒体还应该让人们知道他们洗手时需要洗多长时间、具体步骤如何。一般来说，官方给出的指导越具体、可行，民众越容易理智、冷静地对待危机。
- **每发布一条负面消息，要发布三条正面消息。** 她发现新闻中负面消息比正面消息更多，她表示这是不应该的。她指出在危机中，民众本来就更容易吸收负面消息，因此更应该多发布正面消息。

定制全景图（任务 T6）。教授针对新闻媒体是否对安抚民众采取了行动进行了进一步的探索。这个过程中，她发现大部分共有主题都很具体、内容很明确，不过共有主题 **I** 并不是这样。这个共有主题中的新闻主题和推特主题是因为“virus”和“outbreak”等非常宽泛的关键词而匹配在一起的，实际并不是很相关。因此，她通过匹配修改操作，将这个新闻主题和推特主题分开了。在她修改了这个匹配以后，有一个新的共有主题 **J** 出现了。这个主题中主要是鼓励医生们更加关注流感而不是埃博拉（“Doctors More Concerned With Flu Season Than Ebola Virus”）。教授很高

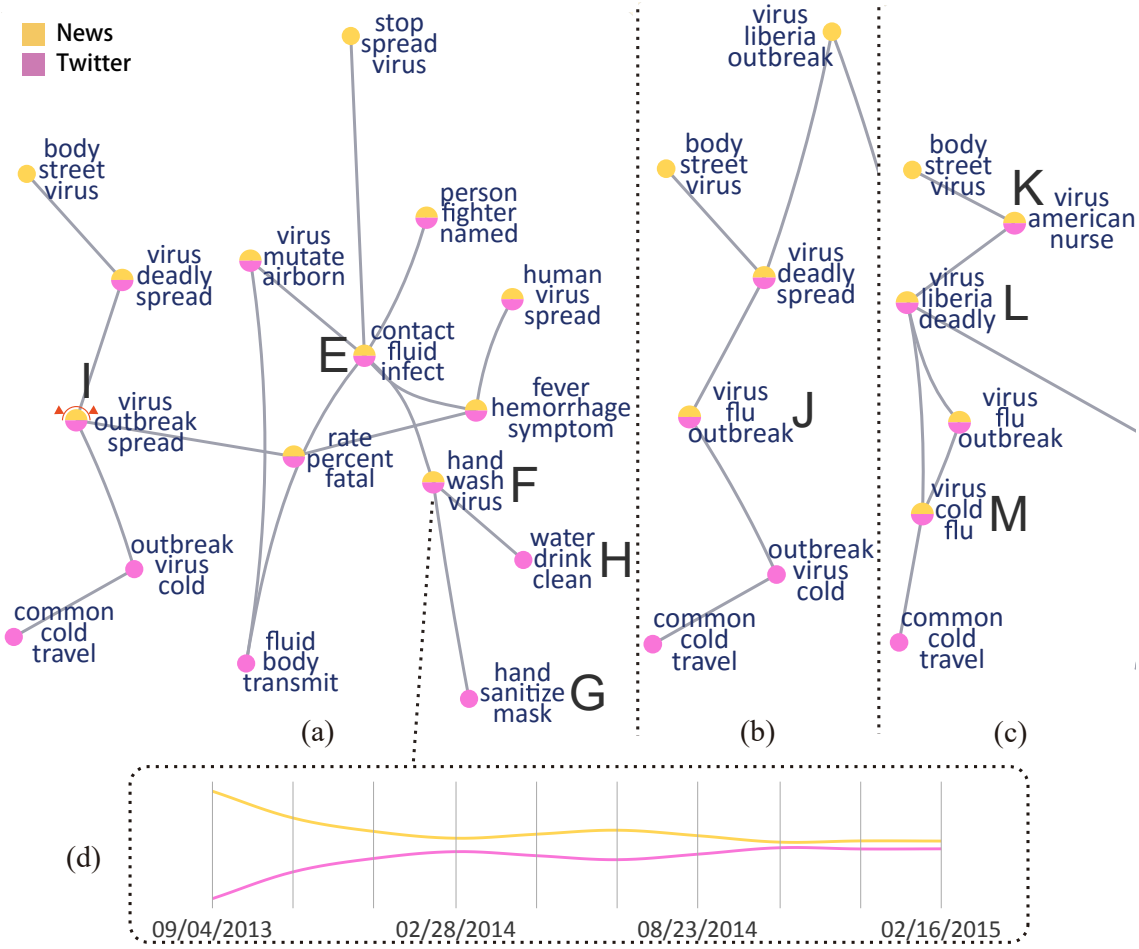


图 4.17　埃博拉病毒相关的知识。(a) 匹配好的主题图；(b) 用户对匹配 **I** 进行修改以后，出现了一个将埃博拉与流感进行对比的共有主题 **J**；(c) 用户对匹配 **J** 进行确认以后，出现了与美国埃博拉病毒相关的主题 **K**、与利比亚埃博拉病毒相关的主题 **L** 以及与感冒相关的主题 **M**；(d) 主题 **F** 的领先-滞后关系。

兴见到这个主题，她表示这个主题说明新闻媒体还是在引导人们更理性地理解埃博拉。因此，她利用匹配修改操作降低了这个主题的不确定性。接着，三个内容具体的共有主题（**K**, **L** 与 **M**）出现了。它们是关于美国埃博拉病毒（**K**）、利比亚埃博拉病毒（**L**）以及感冒（**M**）的主题。这些主题进一步反应出新闻媒体有引导民众更理智地认识埃博拉。例如，主题 **M** 中，一篇新闻的标题是“Gov. Perry: Ebola is much harder to get than the cold”。

在检查完知识相关的主题以后，教授还希望了解一下关于埃博拉中引起轰动的事件。她回忆起有很多跟护士相关的事件，因此她搜索了关键词“nurse”。在这些主题中，与西班牙护士 Kaci Hichox 相关的主题最多，因此她放大到了相关主题（图 4.18(a)）。从主题 **N** 中，她了解到西班牙护士 Kaci Hickox 从西非返回，被政府要求进行为期 21 天的隔离。护士不愿意被隔离，因此起诉了政府。相关推特显示，此时很多人对护士是支持的（“Fight back, Kaci! Nurse Kaci Hickox: ‘I won’t be bullied

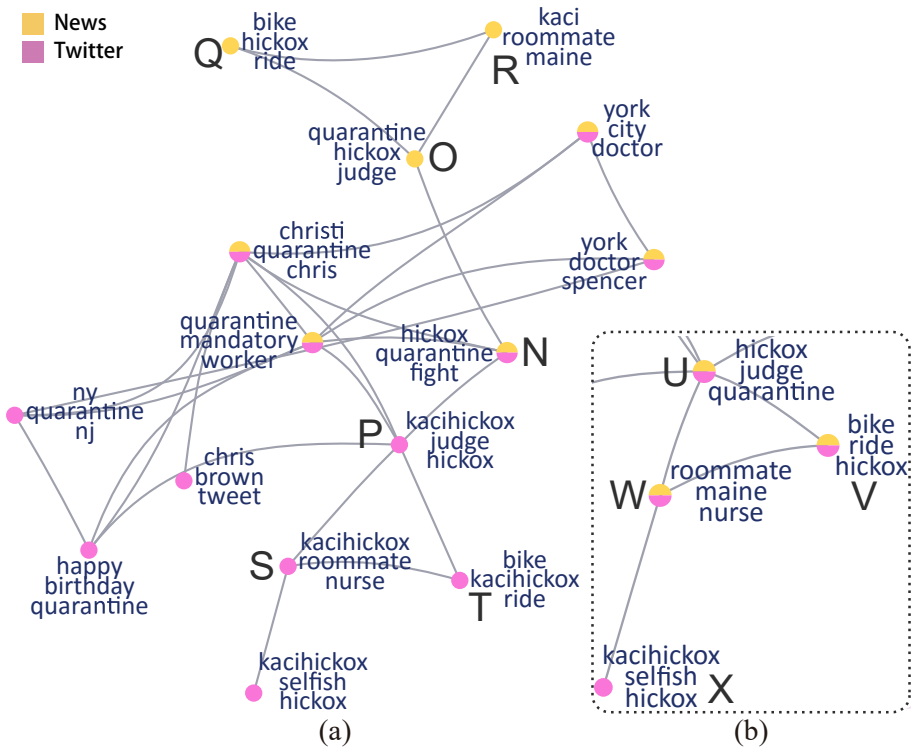


图 4.18　护士 Kaci Hickox 的隔离。(a) 与 Kaci Hickox 相关的主题；(b) 在对主题 **O** 和主题 **P** 进行了匹配以后，三个与护士相关的共有主题（**U**, **V** 与 **W**）出现了。

by politicians'")。为了了解更多信息，教授检查了与 **N** 相关的主题。她发现主题 **O** 和主题 **P** 都是关于法官判定她不应该被隔离（"Maine judge rejects Ebola quarantine for nurse Kaci Hickox")。这两个主题之所以没有被匹配在一起，是因为新闻和推特中提到护士时用语不同。新闻中几乎都是用"Kaci Hickox"，而推特几乎都是用"#kacihickox"。教授匹配这两个主题后，三个新的共有主题 **U**、**V** 和 **W** 出现了。主题 **U** 是由 **O** 和 **P** 匹配而成，主题 **V** 是由 **Q** 和 **T** 匹配而成，而 **W** 是由 **R** 和 **S** 匹配而成。主题 **V** 和 **W** 显示出了媒体发布的关于护士的消极新闻是如何使得公众对于护士看法发生转变的。在主题 **V** 中，新闻媒体报道了 Kaci Hickox 在回家以后，并没有待在家里，而是去外面骑行。在主题 **W** 中，新闻媒体披露 Kaci Hickox 在西非的室友感染上了埃博拉。从这两个主题对应的推特消息中可以看出，民众开始责怪她的自私（"RT @newswatchcanada: #Maine: Selfish #CDC nurse #KaciHickox's roommate in #Africa developed #Ebola")。另外，还有一个与 **W** 相关的推特独有主题专门责备这名护士（**X**, "RT @StevenRosenblum: This is why #KaciHickox is a selfish, unprofessional 'nurse' who should lose her license. #Ebola #Quarantine")。教授评论说尽管美国个人主义盛行，但是在危机发生期间人们更容易有集体主义倾向。她说这也是为什么媒体关于护士的消极新闻很容易改变人们对护士的看法。

4.7.3　专家访谈

我们采访了与我们合作的六个领域专家。采访平均用了 90 分钟。其中包括 10 分钟的系统介绍，50 分钟的案例分析和自由探索，还有 30 分钟的后期访谈。总的来说，专家们都认为 TopicPanorama 很有用。我们将专家访谈的结果总结如下。

图匹配。 所有专家都认为图匹配模块对于他们分析主题全景图很重要。他们尤其喜欢图匹配结果修改算法。一个专家说到：“我很喜欢对全景图的修改功能，这个功能让我可以按照自己的需求修改结果。”

交互式可视化。 专家们对于可视化的作用有了深刻的印象。他们都喜欢混合的可视化形式，这让他们能够从多层级分析主题全景图。另外，他们认为不确定性的符号提供了一个很直观的方式让他们寻找比较可能出错的匹配。

用户使用模式。 用户使用匹配修改操作的频率由特定的用户任务和应用决定。根据我们领域专家的反馈，如果他们只是希望得到一个粗略的概览，通常只需要进行三至五次修改。如果他们需要利用系统完成一个具体、重要的任务（如研究如何在公共健康危机中更好地引导民众），他们可以接受更多次的修改操作。例如，一个专家说如果任务特别重要，他们可以接受进行几十次匹配修改。

4.8　小结及结论

我们通过与一组领域专家的合作，总结出来了一系列的分析任务。基于这些分析任务，我们开发了 TopicPanorama 来帮助用户分析多源文本中的主题全景图。

我们的方法有三个主要贡献。首先，我们提出了一个能够快速地对多张主题图进行一致图匹配的算法。然后，我们设计了一个基于 LOD 的可视化方法。这个可视化设计允许用户从全局概览到局部细节多个层级观察主题全景图。最后，我们开发了交互式图匹配结果修改算法，允许用户生成最符合自己的信息需求的全景图。

我们的方法可能用到其他的工作中。首先，度量学习和特征选择方法可以用于分析其他高维数据。另外，主题树的生成算法也可以应用在其他可视分析系统中，帮助这些系统更好地组织大量数据。

我们的方法仍然有几个不足之处。尽管我们的图匹配算法和可视化方法可以处理四个及以上的文本集合（文本源），但是如果可视化的文本集合太多，在有限空间内会造成较多视觉混乱，影响用户进行分析。现有研究成果表明，人在进行可视化比较时，大约可以跟踪四个物体[100]。另外，我们对专家进行了采访。他们表示在实际应用中，很少对四个文本源同时进行对比分析。因此，TopicPanorama 在很多实际应用中都有用。另外，目前有些主题的提取结果还不甚理想，我们主要采取主题排序的方式来去除掉有噪音的主题。另外一个可能的解决方案是允许用户

交互式地修改主题[116]。

未来的研究方向包括支持主题挖掘结果修改与设计一个适合用在四个及以上个数文档集合的可视化展现形式。

第 5 章　多源动态文本的主题挖掘与可视分析

现在我们面临的是一个高度复杂的信息环境。网络上的普通大众在不断产生和交换各种主题，这些主题在网络上有着不同的生存周期。有的主题传播时间长，影响力大，流动到了各个文本源；有的主题传播时间短，影响力小，只在特定的文本源中流动。跟踪这些主题的流动能够使我们更丰富地认知一个主题如何在网络上出现和发展，能够提高我们对主题的合并与分裂的理解，还能够使得我们更深刻地理解网络上不同文本源中的用户所扮演的角色。

例如，公共卫生研究人员常常会分析一些对大众造成较大危害的疾病（如埃博拉），希望知道如何快速、有效地应对健康危机。通过分析一系列埃博拉相关主题在不同文本源的领先-滞后关系，他们可以发现最具影响力的文本源，并且向政府提出相关建议。领先-滞后关系指的是一个主题在文本源 *A*（领先）中先于另一个主题在文本源 *B*（滞后）中提到。跟踪主题在不同文本源的流动，尤其是它们之间的领先-滞后关系，在学术研究和实际应用中有着越来越大的需求。

跟踪网上大量文本中主题流动情况颇具挑战。我们面临的挑战主要有两个。

第一个挑战是对主题和它们之间的领先-滞后关系随时间的变化进行建模。现有的方法或者只利用文本内容[117]，或者只利用词频的时间序列之间的协整关系[79]对主题的流动进行建模。一般来说，它们建模的准确性不够令人满意。另外，在现实生活中应用时，我们希望能够跟踪主题在两个以上文本源中的流动。因为缺乏一个一致的方法来同时分析两个以上文本源的领先-滞后关系，现有的算法或者跟踪的是相关主题在两个文本源上的流动[79]，或者跟踪的是同一个主题（而不是一

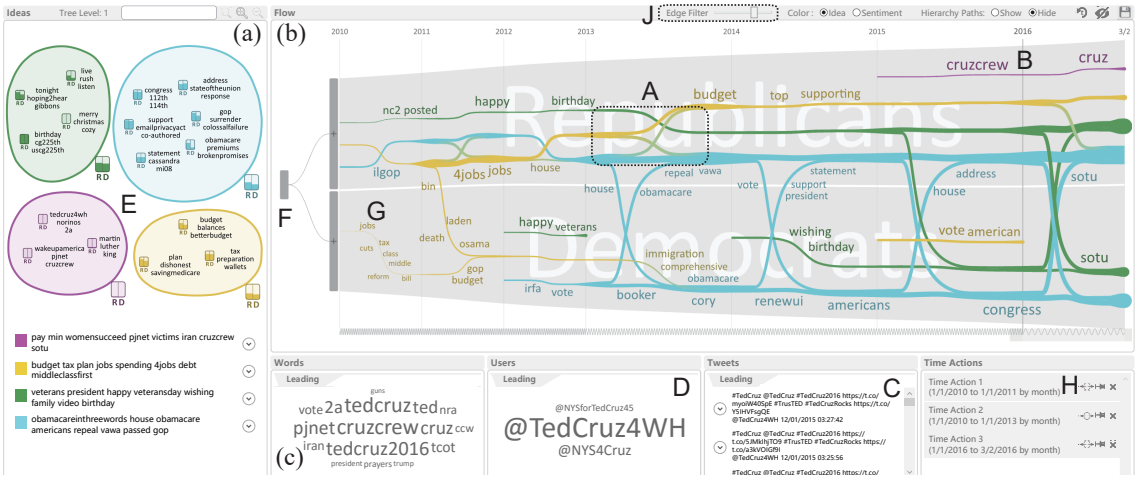


图 5.1　在美国议会数据上的主题以及它们的流动情况的概览。(a) 主题的概览；(b) 主题领先-滞后关系随时间的变化；(c) 信息面板。

系列相关主题）在多个文本源上的流动[117]，主题信息丰富性方面有所不足。

第二个挑战是设计直观的可视化来展示很多主题长时间的流动情况。主题领先-滞后关系模型往往能提取出几百个主题和几千个领先-滞后关系。将它们全部进行展示将会造成视觉混乱与歧义。另外，一个数据集可能含有上千个时间点。因此，我们需要设计一个直观的视觉隐喻，帮助专家快速、有效理解主题长时间的流动情况。这个视觉隐喻需要能够帮助专家在仔细研究感兴趣的时间段时保持对整体上下文的理解。

我们设计了一个可视分析系统来应对上面的挑战。为了能够更准确地对主题流动情况进行建模，我们设计了一个基于随机游走的相关模型，并将它跟贝叶斯条件协整[118] 以及张量分解技术[119] 结合在一起。具体来说，我们首先利用贝叶斯条件协整算法计算词频时间序列之间的相关性。这个相关性被输入到基于随机游走的相关模型来计算词之间随时间变化的相关性。这个相关模型综合考虑了词频的时间序列，推特消息内容，以及元数据（例如转发关系）来更准确地计算随时间变化的词相关性。然后，我们利用基于张量分解的算法来对词进行聚类。每一类词就是一个主题。随时间变化的词的相关性聚合成主题之间的领先-滞后关系。为了展现复杂的领先-滞后关系，我们设计了一个可视界面。这个可视界面结合了基于 Voronoi 树图的气泡树、基于相关聚类的流向图以及焦点加上下文时间轴的优点。基于 Voronoi 树图的气泡树能够帮助用户浏览大量主题。基于相关聚类的流向图算法可以同时对多个流向图进行聚类，从而减少视觉混乱与歧义。焦点加上下文的时间轴能帮助用户浏览长时间流动的主题。

本工作的主要贡献为：

- 一个帮助专家理解、分析多源相关主题之间领先-滞后关系的可视分析系统。
- 一个提高了准确性的基于随机游走的相关模型。
- 一个结合了基于 Voronoi 树图的气泡树、基于相关聚类的流向图以及焦点加上下文时间轴优点的可视化，该可视化可以帮助用户从多个层级来理解大量文本中的主题领先-滞后关系。

5.1　问题分析与系统框架

我们参照 Munzner 等人提出的设计、验证可视系统的嵌套模型[120] 来分析问题、对用户进行调研，从而设计出符合用户需求的系统。具体来说，这个过程分为三个阶段。在第一个阶段，我们对两个领域专家进行了采访，了解他们在社交媒体方面的研究。其中，一个专家研究领域是媒体传播（P1），另一个专家的研究领域是公共卫生（P2）。这些采访平均时长为一个小时。采访过程中，我们了解到专家

在平常研究过程中面临的分析问题、分析过程以及主要挑战。基于这些收集到的信息，我们总结了系统需要满足的需求。在第二个阶段，我们根据总结出来的需求设计、开发了系统原型，并且在一系列对专家的采访中对原型进行了迭代式的完善。这个迭代的过程持续了大概六个月。这个过程中，我们尝试了不同的对领先-滞后关系进行可视化的方法，包括直线、边束化以及流向图。另外，专家们还分别对各个模块进行了评估。在第三个阶段，我们利用系统找出了一些初始的案例，并且将这些案例向专家进行了演示。专家在观看完演示以后，在他们自己的数据集上使用了系统。P1 使用了约两个小时，P2 使用了约一个小时。此后，专家对系统总体的优缺点进行了评价。

5.1.1　分析问题、分析过程以及主要挑战

分析问题。 P1 和 P2 分别研究的是政治传媒和健康危害。在第一阶段的采访中，P1 主要提到了他如何研究美国议会成员之间的讨论和辩论，P2 主要谈到她在埃博拉疫情方面的研究。这两个专家都声称他们的研究中，常常涉及到各种社会议题和它们之间的交互，他们往往从人物（who）、事件（what）以及时间（when）等几个角度来研究这些议题和它们的变化。P1 提到 who、what 和 when 是传媒研究中常常涉及的 Lasswellian 问题[121]。详细的分析问题在 5.1.2节中会和相应需求一起介绍。

分析过程和主要挑战。 在专家的研究中，他们常常要从无条理的推特数据中提取相关的社会议题，并且研究多个社会议题之间如何相互影响，以及影响如何随时间变化。另外，他们对于经常发起新议题并且影响了讨论进程的用户与机构非常感兴趣。为此，他们在目前的研究中常常通过对关键词进行搜索来找到相关的推特消息，并且手动将得到的文本按照用户分为不同的文本源，然后对大量推特消息进行阅读，总结其中包含的社会议题，并且跟踪这些社会议题之间的相互影响。这个过程需要大量的人力，而且一些重要的信息也可能在这个耗时耗力的过程中丢失。专家们同意议题和我们提到的主题之间有着直接的对应关系。他们也同意主题的之间的领先-滞后关系体现了社会议题之间的交互和影响。

5.1.2　设计需求

通过总结采访中获取的信息，我们得到了下面的系统需求。

R1. **生成主题概览。** 专家表示他们在研究初期常常要将无条理的原始数据转化成有条理的、意义明确的概览。这个概览包含了对主题的描述以及它们之间的全局领先-滞后关系。这里，全局领先-滞后指的是该主题在所有时间点上的综合的领先-滞后情况。特别的，我们发现专家有如下的需求：

R1.1. **总结主题与主题在不同文本源上的全局领先-滞后关系。** 具体来说，系统需要能够自动提取有意义的主题以及它们的全局领先-滞后关系，并且将这个信息以一种组织清晰的方式呈现给领域专家。这个需求是从专家们的分析问题中总结出的。相关的分析问题的例子是“数据中大概有多少类主题？这些主题的主要关注点是什么？哪个文本源在这些主题上处于领先地位？领先-滞后关系的总体规律是什么？”回答这些问题有助于专家们加深对数据的理解，并且快速发现感兴趣的主题和领先-滞后关系。

R1.2. **从多个层级观察主题。** 专家提到在他们目前的研究过程中，经常从多个层级分析主题。P1 提到如果系统只呈现抽象的主题（例如经济相关主题），它的理论价值将较为局限。他们往往更加关心抽象主题包含的具体的子主题，例如医疗保险等。但是在他们对数据进行分析的过程中，往往更喜欢从抽象主题入手，先了解数据中包含哪些抽象主题以及这些主题之间的影响，然后再将这些抽象的规律进行分解，进一步研究具体的主题。

R2. **跟踪主题的流动。** 在专家们的研究中，一个主要的关注点是主题之间如何互相影响。相应的，我们发现他们在研究相关主题之间交互方面有着如下的需求：

R2.1 **跟踪相关主题在不同文本源上的领先-滞后关系随时间的变化。**在专家的研究过程中，他们常常研究的一些问题包括：“哪些文本源在不同的时间段（例如周末、工作日、竞选投票期以及疫情反复期）最具影响力？这些领先-滞后关系在竞选（或者埃博拉疫情的）的不同阶段是会发生变化，还是会维持相对稳定？”

R2.2 **探索、比较主题在不同层级上的流动情况。**首先，跟踪、比较局部领先-滞后关系随时间的变化回答了 Lasswellian 问题中“when”相关的问题，使得专家可以先发现较长时间片段上的领先-滞后关系，然后再将在这个较长时间片段上发现的规律进行细化，研究较为具体的时间片段（例如以周或者天为单位）上的规律。

然后，在很多应用中，文本源是自然地组织成层次结构的。相应的，这些领域专家需要比较不同层级文本源上主题流动的情况。例如，P1 把不同用户群体发的推特消息分成不同的文本源。他说到：“在我现在的研究中，我需要比较民主党的推特消息与共和党的推特消息之间的领先-滞后关系。在每个党内，我又希望了解参议院成员和众议院成员间的推特消息之间的领先-滞后关系。”

R2.3 **发现有影响力的文本源。** 发现有影响力的文本源对于两个专家的研究都非常重要。这使得他们可以研究不同文本源的社会影响力以及文本源上用户的沟通策略。专家们研究的主要分析问题是：“在这个主题上，哪个文本源在这段时间处于领先的位置？总体来说，哪个文本源更加有影响力？”

R3. **探索具体信息。**领域专家们在描述他们的分析过程时，都强调了能够分析、研

究具体信息的重要性。分析具体的信息对于发现规律以及验证假设都很重要。相应的，系统需要支持：

R3.1 探索关键内容。为了理解一个主题或者主题之间的领先-滞后关系，专家们需要能够观察、分析一些关键内容，包括相关的推特消息和关键词。这个需求是从专家们以下的分析问题中总结出来的：“有没有一些推特消息能够帮助解释这个领先-滞后关系？哪些关键词与这个领先-滞后关系的变化有关？”

R3.2 发现关键的用户，并且理解他们所扮演的角色。专家表示，找出造成领先-滞后关系变化的关键用户非常重要。这可以帮助政策制订者和业界从业人员（例如公共关系专家）在合适的时间点采取合适的行动来控制或者加速这样的变化。相关的分析问题包括：“谁在这个主题上起到了关键作用？这个人在这个领先-滞后关系上扮演着什么角色？”

5.1.3　系统框架

基于上述需求，我们设计、开发了一个可视分析系统。如图 5.2所示，我们的系统包括三个主要的模块，即领先-滞后分析模块、层次化聚类模块，以及领先-滞后关系可视化模块。给定一些推特消息和相应的元数据，领先-滞后分析模块提取出主题和它们在不同文本源上的领先-滞后关系随时间的变化。为了能够使得用户可以从多个层级观察主题和它们之间的领先-滞后关系，我们利用层次化聚类模块来对主题、时间以及文本源进行层次聚类。层次聚类的结果、主题和主题间领先-滞后关系被送到领先-滞后可视化模块进行展现。可视化模块包含三个部分。第一部

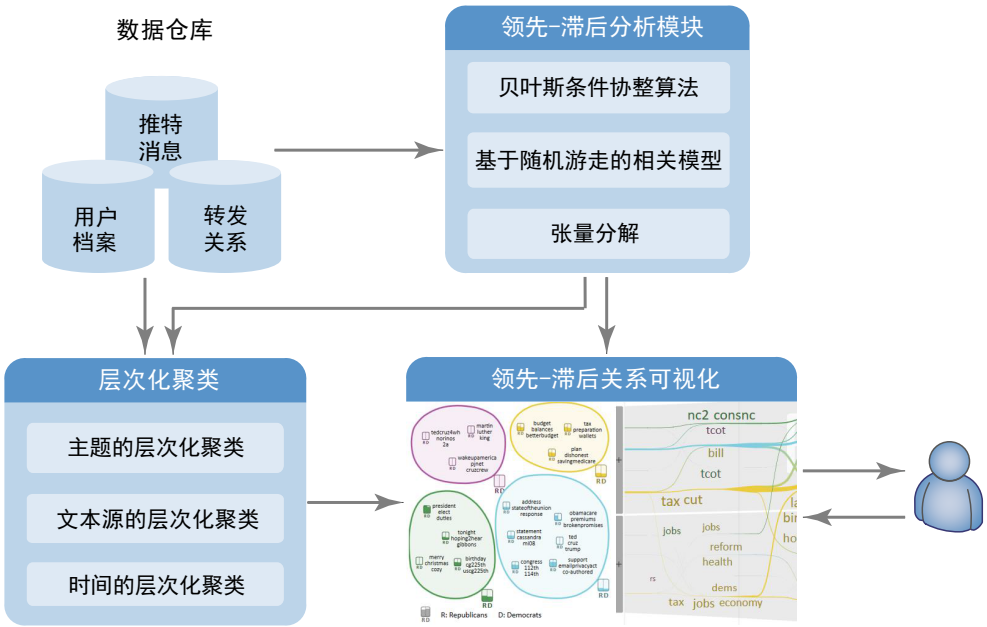


图 5.2　系统框架。

分是气泡树。它展示的是主题和主题的层次结构，为专家提供主题的概览（R1.1、R1.2）。第二个部分是基于相关聚类的流向图，它展示的是相关主题的领先-滞后关系如何随时间发展、变化（R2.1、R2.3）。我们将这个流向图与焦点加上下文的时间轴相结合，使得用户可以在不同的时间粒度（年、月、周等）上对领先-滞后关系进行探索（R2.2）。系统还提供了一些交互技术，帮助用户探索、比较细节信息（R3.1、R3.2）。

5.2　主题和领先-滞后关系挖掘

这节中，我们将介绍提出的挖掘主题之间领先-滞后关系的算法。

5.2.1　主要思想

我们的算法主要包含两步。

- **增强词图的构建。** 在这一步中，我们通过提取词之间随时间变化的相关性来构建增强词图。如图 5.3(a) 所示，增强词图中每个顶点表示一个文本源中的一个单词。与普通的图相比，增强图中的边代表了词之间随时间变化的相关性，这个相关性由一个相关性向量 $\mathbf{c} = [c_1, ..., c_T]$ 和一个领先-滞后向量 $\mathbf{\Delta t} = [\Delta t_1, ..., \Delta t_T]$ 表示。
- **增强词图的分割。** 如图 5.3(b) 所示，在这一步中，主题是通过对增强词图进行分割来计算的。我们将词聚合成主题，将词之间随时间变化的相关性聚合成主题之间的领先-滞后关系。

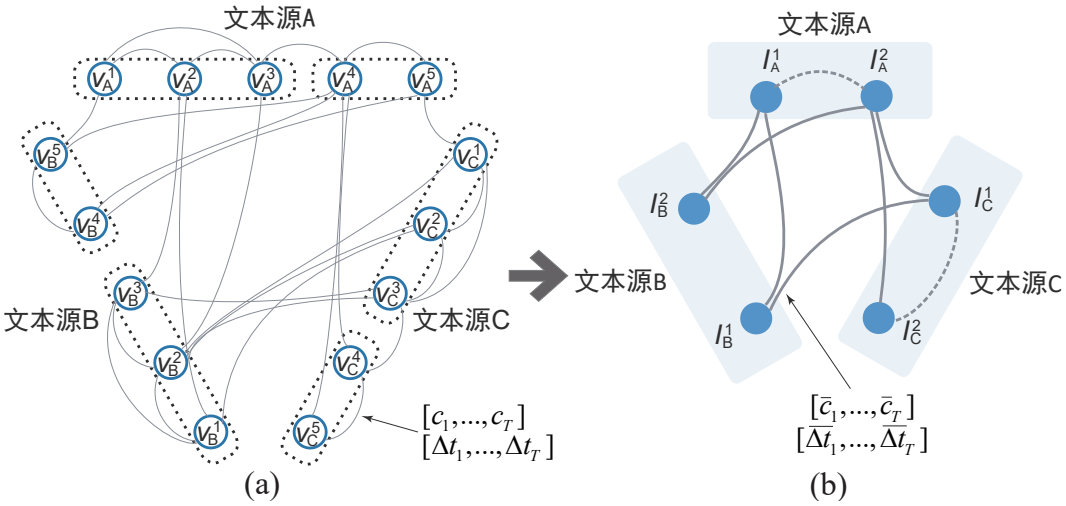


图 5.3　主题领先-滞后关系的挖掘算法：(a) 增强词图；(b) 主题之间的领先-滞后关系。

5.2.2　增强词图的构建

给定 N 个文本源和 M 个词，增强词图中的顶点 v_n^m （$1 \le n \le N, 1 \le m \le M$）代表第 n 个文本源中的第 m 个词。增强词图构建的主要目标是准确提取顶点之间随时间变化的相关性。如图 5.3所示，随时间变化的相关性由一个相关性向量 $\mathbf{c} = [c_1, ..., c_T]$ 和一个领先-滞后向量 $\mathbf{\Delta t} = [\Delta t_1, ..., \Delta t_T]$ 表示。这里，T 代表的是时间点的个数，c_k 代表的是第 k 个时间点顶点之间的相关性，Δt_k 代表的是第 k 个时间点两个顶点之间的领先-滞后时间差。

这里，主要的难点在于如何准确地计算词之间随时间变化的相关性。现有最先进的方法[79] 通过计算词频时间序列之间的协整关系来计算词的相关性。这个方法的主要问题是它忽略了文本的内容以及其他元数据，例如用户之间的关注关系等。因此，这个算法的准确性并不是非常令人满意。为了解决这个问题，我们开发了一个基于随机游走的相关模型。这个模型同时考虑了推特消息的内容、元数据以及词之间的协整关系。

我们的模型是受到信息检索领域的三层互增强模型[122] 的启发设计的。如图 5.4所示，这个相关模型包含三个图（用户图，推特消息图，词图）以及这三个图之间的连边。用户图是根据用户之间的关注关系建立的。推特消息图是根据推特消息转发关系建立的。词图是根据词频时间序列之间的协整关系建立的。具体来说，我们提取每个词的词频时间序列，两个词之间的每个时间点上的协整关系通过贝叶斯条件协整算法[118] 计算。两个图之间的连边是通过推特消息内容和发布关系来生成的。具体来说，我们将每个用户与他/她发布的所有推特消息以及提到的所有词连在一起。另外，每条推特消息跟它里面包含的词连接在一起。在每个时刻 k，我们根据 $[k, k+\tau]$ 时刻发布的推特消息来建立相关模型。这里 τ 是允许的最大领先-滞后时间差。

接下来，我们介绍如何根据第 k 时刻的相关模型来计算 c_k 和 Δt_k。假设我们需要计算的是顶点 v_A^1 和 v_B^1 之间的 c_k 以及 Δt_k。如果 v_A^1 和 v_B^1 在第 k 时刻相关，那

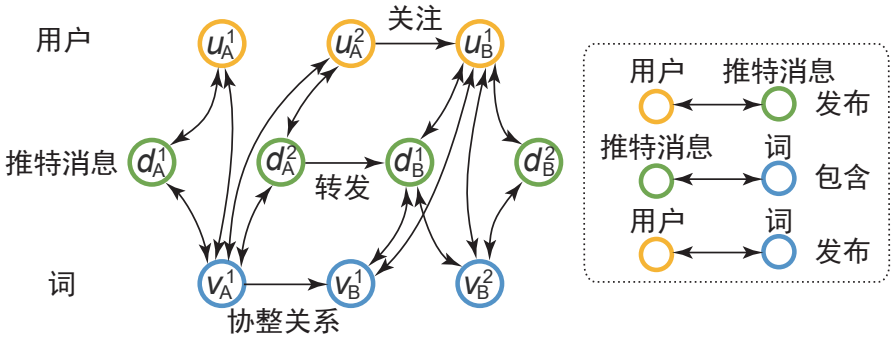


图 5.4　基于随机游走的三层相关模型。

么从 v_A^1 出发的随机游走路径中，应该有一些较短的路径可以从 v_A^1 到达 v_B^1[123]。基于这个思想，我们设计了如下的分为三步的算法。

首先，我们从 v_A^1 出发，采样出一系列随机游走路径。一条随机游走路径由一系列的随机漫步构成。在每次随机漫步时，这个随机游走以 p_s 的概率停止。如果不停止，那么将从邻居节点中挑选一个。这个邻居节点可以是个用户节点，词节点，或者推特消息节点。挑选每个节点的概率与连边权重成正比。在图 5.4中，一个可能的随机游走路径是 $\{v_A^1 \to d_A^2 \to d_B^1 \to v_B^1\}$。

然后，我们过滤掉一些无效的随机游走路径。在我们的应用中，并不是所有随机游走路径都有效。例如，如果 d_A^2 在 d_B^1 之后发布，那么 d_A^2 中包含的信息无法流动到 d_B^1。在这种情况下，$\{v_A^1 \to d_A^2 \to d_B^1 \to v_B^1\}$ 是一条无效路径。具体来说，当且仅当 $t(d^{(l_1)}) \le t(d^{(l_2)}) \le ... \le t(d^{(l_Q)})$ 时，我们称一个随机游走路径 l 为有效的。这里，Q 是 l 中推特消息的条数，$d^{(l_q)}(1 \le q \le Q)$ 是 l 中的第 q 条推特消息，$t(d^{(l_q)})$ 表示 $d^{(l_q)}$ 发布的时间。

最后，我们根据有效的随机游走路径来计算 c_k 以及 Δt_k。具体来说，c_k 是从 v_A^1 出发的有效的随机游走路径到达 v_B^1 的经验概率：

$$c_k = |\mathcal{L}(v_A^1 \to v_B^1)|/|\mathcal{L}(v_A^1)| \tag{5-1}$$

这里 $\mathcal{L}(v_A^1)$ 是从 v_A^1 出发的所有有效随机游走路径的集合，$\mathcal{L}(v_A^1 \to v_B^1)$ 是从 v_A^1 出发，到达 v_B^1 的所有有效随机游走路径的集合，$|\cdot|$ 代表集合中的元素个数。接下来，我们通过把所有随机游走路径中推算出来的领先-滞后时间进行平均来计算 Δt_k：

$$\Delta t_k = \frac{\sum_{l \in \mathcal{L}(v_A^1 \to v_B^1)} \left[t(d^{(l_{Q'})}) - t(d^{(l_1)}) \right]}{|\mathcal{L}(v_A^1 \to v_B^1)|} \tag{5-2}$$

这里 $d^{(l_{Q'})}$ 是随机游走路径 l 中在到达 v_B^1 之前经过的最后一个推特消息节点，$d^{(l_1)}$ 是随机游走路径 l 上的第一个推特消息节点。

5.2.3　增强词图的分割

要准确提取主题和主题之间的领先-滞后关系，一个主要的难点在于有效地对增强词图进行分割。传统的图分割的方法不能直接应用于增强图，因为增强图中的每条边是由两个向量，而不是一个实数值来表示的。Zhong 等人[79] 提出了一个基于张量的算法来对增强二部图进行分割，但是这种算法只能对两个文本源之间的领先-滞后关系进行建模。为了解决这个问题，我们用一个六维张量来对多个文

本源上主题之间的领先-滞后关系进行建模。

如图 5.3(a) 所示，增强词图由来自不同文本源的词和它们之间的随时间变化的相关性组成。其中，随时间变化的相关性由两个向量表示。为了将这个多维的数据用一个统一的模型表示，我们将 Zhong 等人[79] 提出的四维张量拓展为一个六维张量 $\mathbf{X} \in \mathbb{R}^{N\times M\times N\times M\times T\times(\tau+1)}$。在这个张量中，前四维分别对两个节点之间的边进行了编码，第五维对时间进行了编码，第六维对领先-滞后时间差进行了编码。具体来说，如果在第 k 个时间点顶点 v_n^m 和 $v_{n'}^{m'}$ 之间满足 $\Delta t_k = h$，我们就将 $\mathbf{X}_{nmn'm'kh}$ 设为 c_k。否则，我们将 $\mathbf{X}_{nmn'm'kh}$ 设为 0。

接着，我们利用 Zhong 等人[79] 提出的张量分解方法来提取每个词的特征向量。我们采用的张量分解方法是贪心算法 PARAFAC[119]，它可以提取一个秩为 q 的 $\mathbf{X}$ 的近似：

$$\mathbf{X} \approx \sum_{i=1}^{q} \lambda^{(i)}\mathbf{a}^{(i)} \circ \mathbf{r}^{(i)} \circ \mathbf{b}^{(i)} \circ \mathbf{s}^{(i)} \circ \mathbf{x}^{(i)} \circ \mathbf{y}^{(i)} \tag{5-3}$$

这里 $\lambda^{(i)} \in \mathbb{R}$，$\mathbf{a}^{(i)} \in \mathbb{R}^N$，$\mathbf{r}^{(i)} \in \mathbb{R}^M$，$\mathbf{b}^{(i)} \in \mathbb{R}^N$，$\mathbf{s}^{(i)} \in \mathbb{R}^M$，$\mathbf{x}^{(i)} \in \mathbb{R}^T$，$\mathbf{y}^{(i)} \in \mathbb{R}^{(\tau+1)}$。$\mathbf{a}^{(i)}\circ\mathbf{r}^{(i)}\circ\mathbf{b}^{(i)}\circ\mathbf{s}^{(i)}\circ\mathbf{x}^{(i)}\circ\mathbf{y}^{(i)}$ 是六路外积。具体来说，$(\mathbf{a}^{(i)}\circ\mathbf{r}^{(i)}\circ\mathbf{b}^{(i)}\circ\mathbf{s}^{(i)}\circ\mathbf{x}^{(i)}\circ\mathbf{y}^{(i)})_{nmn'm'kh} = \mathbf{a}_n^{(i)}\mathbf{r}_m^{(i)}\mathbf{b}_{n'}^{(i)}\mathbf{s}_{m'}^{(i)}\mathbf{x}_k^{(i)}\mathbf{y}_h^{(i)}$。这里 $\mathbf{a}_n^{(i)}$ 是向量 $\mathbf{a}^{(i)}$ 的第 n 维的取值。

然后，我们利用因子 $\{\mathbf{r}^{(i)}\}$ 以及 $\{\mathbf{s}^{(i)}\}$ 来计算每个词的特征向量。具体来说，我们构造了一个特征矩阵 $\mathbf{S}$：

$$\mathbf{S} = \left[\mathbf{r}^{(1)}, ..., \mathbf{r}^{(q)}, \mathbf{s}^{(1)}, ..., \mathbf{s}^{(q)}\right] \tag{5-4}$$

对于第 m 个词，我们将它的特征向量设为 $[S_{m1}, S_{m2}, ..., S_{m(2q)}]$。

最后，我们利用非负矩阵分解（NMF，Non-negative Matrix Factorization）[124] 来对词进行聚类。每类词就形成了一个主题。两个主题之间的相关性是通过将词之间的相关性相加来得到的：$\bar{\mathbf{c}} = \sum \mathbf{c}$。两个主题之间在 k 时刻的领先-滞后时间是通过将词之间的领先-滞后时间差进行平均得到的：$\overline{\Delta t_k} = \sum(c_k\Delta t_k)/\sum c_k$。

5.3　可视化

我们根据第 5.1.2节中讨论的系统需求设计了我们的可视化界面。该可视化界面主要由三个部分组成：主题可视化、领先-滞后关系可视化以及信息面板。信息面板主要用于帮助用户更方便地探索文本源的细节信息（R3）。用户点击一个主题

流以后，可以在信息面板看到主题流对应的关键内容（R3.1）以及起主导作用的用户（R3.2）。下面，我们重点介绍主题可视化与领先-滞后关系可视化。

5.3.1　主题可视化

为了总结大量的主题（R1.1）并且让用户可以从多个层级自由探索主题（R1.2），我们利用贝叶斯多分枝树模型[86]将主题组织成一棵多分枝树。对于多分枝树上的每个主题或者主题类，我们分别提取它们的全局领先-滞后关系（R1.2）。接着，我们介绍设计的主题多分枝树与全局领先-滞后关系可视化方法。

5.3.1.1　用柱形符号表示全局领先-滞后

根据从采访中收集到的信息，对于每一个主题，我们将显示它的关键词，让用户可以快速了解这个主题的主要内容。为了让用户能够快速将主题和它的全局领先-滞后关系对应起来，我们将全局领先-滞后的信息显示在每个主题的关键词附近[125]。受到迷你图（Sparkline Visualization）[126]的启示，我们用一个与关键词大小相当的符号来将全局领先-滞后关系与主题关键词和谐地展示在一起。

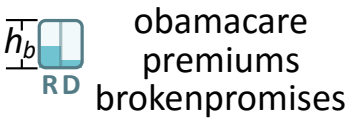


图 5.5　全局领先-滞后符号。

我们的设计如图 5.5所示。在这个设计中，每个文本源用一个与关键词大小相当的迷你柱形图表示。柱形图的填色部分高度代表相应文本源领先的时间比例。具体来说，填色部分的高度为 $h_b T_l/T$。这里 h_b 是柱形图高度（图 5.5），T_l 是相应文本源在特定主题上领先的时间点个数，T 是数据集中的所有时间点个数。

5.3.1.2　用基于 Voronoi 树图的气泡树表示主题树

专家对于主题类（R1.2）以及类别中的重要主题（R1.1）都很感兴趣。因此我们对于每个主题类挑选了一些代表性的主题，并且将这些代表性的主题的关键词和全局领先-滞后关系展示在相应的主题类中。这些代表性的主题是利用 TIARA[17]中提出的主题排序技术进行挑选的。为了能够用较多的代表主题来总结一个主题类的内容，我们需要设计一个空间利用率较高的主题树可视化方法。一个直接的想法是用空间填充的可视化技术例如 Voronoi 树图[111]。但是，因为 Voronoi 树图的边界往往很不规则，直接用 Voronoi 树图展示主题树会导致全局领先-滞后符号

难以进行布局。另外，Voronoi 树图的边界常常并不平滑，导致主题类的可读性降低[127]。为了解决这些问题，我们将 Voronoi 树图与 EulerSmooth[127] 相结合，从而生成一个空间利用率较好的气泡树。气泡树中，每个主题类用一个特定颜色的光滑闭合曲线表示。代表性的主题放置在主题类边界的内部。图 5.1(a) 中上方显示的是气泡树，下方显示的是气泡树上每个主题类的关键词。

布局。 为了对气泡树进行布局，我们首先用 Voronoi 树图来对用户选定层级的主题树进行布局，生成主题类和代表性主题的紧凑的布局结果。然后我们将 Voronoi 树图的布局结果用 EulerSmooth 进行改进。EulerSmooth 是基于力导向的改善边界形状的算法。我们之所以选择 EulerSmooth，是因为：

- 它能保证主题（用关键词和全局领先-滞后关系符号表示）不会彼此重叠；
- 它能生成平滑美观的主题类的边界，提高主题类可读性。

在用 EulerSmooth 对边界进行优化以后，我们将主题类的全局领先-滞后符号放置在主题类闭合曲线旁边，然后再一次利用 EulerSmooth 对布局结果进行优化。

交互。 气泡树使得用户可以从多个层级自由探索主题（R1.2）。用户可以选择一个主题类进行放大，或者点击缩小按钮（图 5.1E）来返回较上层的主题。为了让用户在放大/缩小的过程中更好地跟踪感兴趣的主题，我们利用分阶段的动画[128] 来展示放大/缩小的过程。我们还利用多边形裁剪算法[129] 来在动画过程中更好地展示一个主题类如何分裂成多个子类，以及多个子类如何合并成一个主题类。

5.3.2　领先-滞后关系可视化

图 5.1(b) 展示的是用于显示局部领先-滞后关系的流视图。这个视图把多种可视化技术结合在一起，从而满足领域专家的分析需求（第 5.1.2节）。具体来说，我们将多层级的文本源展示成一个条带树（Stripe Tree，R2.2，R2.3），把局部领先-滞后展示成一个流向图（R2.1）。接下来，我们具体介绍我们的可视化设计、布局算法以及交互。

5.3.2.1　用条带树表示文本源及其层次结构

图 5.1(b) 显示的是条带树。它由一个堆叠树[130] 和一些文本源的条带组成。堆叠树上的每个节点代表一个文本源。对于堆叠树上每个叶子节点对应的文本源，我们用一个条带来表示它随时间变化的活跃程度。这里，活跃程度用相应时间点上该文本源中对应的推特消息条数来表示。

在第一个版本的原型系统中，我们将每个文本源的名字（例如“新闻”）放置在每个条带的左边。当我们将这个信息展示给 P1 以后，他表示在观察主题流动

情况时辨别文本源名字有些困难。他说在试图理解主题流时，他的视线需要不停在左边的文本源名字和右边的主题流之间切换。一个更大的问题是他可能将文本源名字和主题流对应的关键词混在一起。在他的建议下，我们实现了一个用水印（图 5.1G）显示文本源名字的版本。我们将两个版本展示给了 12 位用户。他们其中的 9 位更倾向于用水印这种设计。所有人都同意因为白色水印和主题流的关键词之间有足够的对比度，这个水印不会影响他们对主题流内容的分析。因此，我们在最终的版本中采用了水印这种设计。

交互。堆叠树使得用户可以从多个层级自由探索不同的文本源（R2.2.）。用户可以点击堆叠树上的一个叶子节点来观察更底层的文本源（放大），也可以点击堆叠树上的一个中间节点来回到更上层的文本源（缩小）。我们用分阶段的动画来对这个放大/缩小的过程进行平滑，使得用户能够在这个过程中更好地追踪感兴趣的主题流。

5.3.2.2　用流向图表示局部领先-滞后关系

从数学上说，主题之间的局部领先-滞后关系可以用一个有向无环图（Directed Acyclic Graph，DAG）来表示。如图 5.6(a) 所示，DAG 中的每个节点表示特定时间点来自于某个文本源的一个主题。我们利用 Xu 等人的算法[22] 来对这些节点进行纵向排序。节点的颜色代表它对应的主题类别。如果一个节点领先另一个节点，那么它们之间就会用一个有向边相连。通过这样的方式，这个 DAG 不仅对同一个文本源内部主题的领先-滞后关系进行了编码（例如边 $\overrightarrow{AB}$），还对不同文本源的主题之间的领先-滞后（例如 $\overrightarrow{AC}$）以及领先-滞后时间差进行了编码。

要展示这个 DAG，一个直接的想法是利用桑基图[131]。桑基图是被广泛用来对带权信息流进行可视化的方法。但是，它在显示多个类别之间的信息流时可能造成较大的视觉混乱[130]。另一个选择是用边束化方法[132] 来将流进行捆绑，从而降低视觉混乱。但是，边束化的方法可能造成很多交叉，引入不必要的视觉混乱[133]。为了解决这个问题，我们利用流向图来对 DAG 进行可视化。这是因为流向图可以快速、平滑地将边合在一起，从而有效地减少视觉混乱[133]。

布局。 流向图是用来分析一些物体如何从一个源地点挪动到多个目标地点的[134]。每个流向图包括一个源节点和多个目标节点。因为在 DAG 中有很多源节点（即有着出边的节点），我们需要对多个流向图进行布局，这有可能导致视觉混乱以及歧义。为了更有效地对多个流向图进行布局并且避免歧义，我们开发了一个基于相关聚类的流向图布局算法。这个算法主要包含下面的几个步骤。

算法的第一步是**基于歧义检测的边过滤**。在这个步骤中，我们通过过滤掉不太重要但是造成了很多歧义的边来减少 DAG 中的歧义。这里，边的重要性是用边

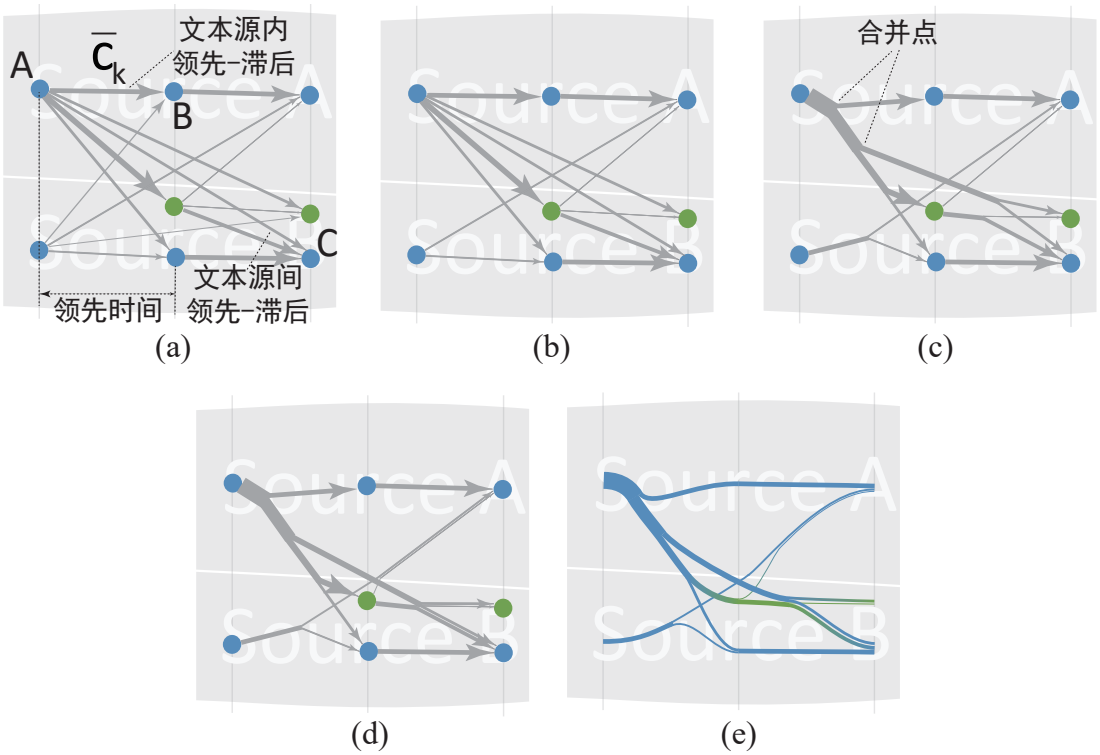


图 5.6　基于相关聚类的流向图布局算法：(a) 将局部领先-滞后关系表示为 DAG；(b) 应用了基于歧义检测的过滤算法以后的 DAG；(c) 将每个源节点单独计算流向图得到的结果；(d) 利用相关聚类同时计算所有流向图的结果；(e) 平滑以后生成的最终流向图。

的权重 $\bar{c}_k$ 来衡量的。我们利用 AmbiguityVis[135] 中的四种衡量标准来计算歧义的大小。这四种衡量标准是点的重叠（g_k^o）、边的交叉（g_k^c）、边交叉的角度（g_k^a）以及点边重叠（g_k^n）。对于每条边，我们用这四个衡量标准来计算它对应的歧义值 g_k。具体来说，g_k 定义为 $\mu_o g_k^o + \mu_c g_k^c + \mu_a g_k^a + \mu_n g_k^n$。这里 μ_o、μ_c、μ_a 以及 μ_n 是用来平衡四种衡量标准的参数。在我们的系统中，我们将 μ_o、μ_c 以及 μ_n 设置为 1，将 μ_a 设置为 0.01。如果 $\bar{c}_k \leq \gamma g_k$，我们就过滤掉第 k 条边。这里 $\bar{c}_k$ 是第 k 条边的权重，γ 是用户提供的用于平衡歧义量和信息量的参数。缺省的，我们将 γ 的值设为 0.02，因为我们发现这个值在很多情况下都能生成较为不错的结果。我们同时还允许用户利用一个滑条（图 5.1J）来根据他们的信息需求自由调整 γ 的取值。图 5.6(b) 显示了在应用基于歧义检测的边过滤以后的 DAG。

算法的第二步是**相关聚类**。在这一步中，我们计算流向图的合并点（Joint Node）。合并点是边进行合并的位置（图 5.6(c)）。现有最先进的方法[133] 是通过将目的点聚类成一棵螺旋树（Spiral Tree）来计算合并点的。当有多个源节点的时候，这个方法是将每个源节点对应的目的节点分别进行聚类。如图 5.6所示，尽管这个做法可以降低 DAG 的视觉混乱，它有可能会无法把不同源节点的相邻边进行合并，导致不必要的视觉混乱。为了解决这个问题，我们利用相关聚类[136] 来同时

建立多棵螺旋树。如图 5.6(d)，我们的算法可以检测出来属于不同流向图的相似子结构，并且把相应的边进行融合，从而减少视觉混乱。

算法的最后一步是**平滑**。在这一步中，我们利用三次贝塞尔曲线将合并点和目的节点平滑地连接在一起。在连接这些节点的时候，我们移动控制点的位置，保证在节点和合并点处的出边和入边的导数方向相同。平滑以后的结果如图 5.6(e) 所示。

5.3.2.3　用弹簧表示焦点加上下文时间轴

条带树和流向图跟一个多焦点的时间轴耦合在一起。这个时间轴可以方便用户从多时间粒度自由探索主题流（R2.2）。受 TextFlow[1] 中通过频率来展示联系紧密程度的启发，我们利用一个弹簧隐喻来有效展示时间被压缩的程度。具体来说，时间轴的弹簧频率用来表示压缩的时间点个数，频率越高，压缩时间点个数越多（图 5.1(b)）。

交互。用户可以通过点击时间轴来选择一个时间区域，通过对时间区域的左侧或者右侧进行拖拽来对这块时间区域进行放大或者缩小。在拖拽的过程中，主题流和文本源条带都会实时更新。如果有足够的空间，选中的时间区域会分裂成粒度更细的多个时间区域。用户选择、放大、缩小时间区域的操作会自动记录在时间轴操作记录面板（图 5.1H）上。通过这个面板，用户可以对时间区域进行标注，还可以轻松回到以前选中的时间区域（R2.2）。

这四个部分分别是词面板、推特消息面板、用户面板以及时间轴操作记录面板。用户点击一个主题流以后，可以在词面板和推特消息面板看到主题流对应的关键内容（R3.1），还可以在用户面板看到起主导作用的用户（R3.2）。

5.4　数值实验

在这个实验中，我们证明考虑了推特消息内容、元数据以及时间序列之间的协整关系以后，模型准确性可以有显著地提高。

5.4.1　数据集

在实验中我们用到了下面两个数据集。

- **政党数据集 A**。这个数据集包含第 114 届美国议会成员的 1,102 个推特账号发布的 1,605,361 条推特消息。这些推特消息的时间区间是从 2010 年 1 月到 2016 年 3 月。这些账号是由一个传播学硕士手动查找并且标注的。根据专家的需求，我们将这些账号分成了四个用户群体：民主党的众议院成员、参议院成员以及共和党的众议院成员、参议院成员，并将不同用户群体所发的推

特信息认为是不同的文本源。我们利用 Tweetinvi[137] 来抓取这些账号之间的关注关系。

- **埃博拉数据集 B**。这个数据集包含 321,114 个推特账号在 2014 年 7 月到 2016 年 2 月期间发布的 16,711,670 条与埃博拉相关的推特消息。我们利用 Tweetinvi 来抓取这些账号的档案以及他们之间的关注关系。根据专家的需求，我们根据位置信息将这些推特消息划分为不同文本源。这些位置信息是使用由 Full Contact 提供的 API 来进行去噪和组织的。

我们利用以下三步来处理这些数据集。首先，我们将这些推特消息按照时间进行分组，每组包含某天的推特消息。我们在数据集 A 上总共得到 2,253 组（时间点），数据集 B 上总共得到 574 组（时间点）。然后，我们对于每个时间点 k 建立了一个相关模型（图 5.4）。这个模型是利用第 k 组到第 $k+\tau$ 组的推特消息建立的。我们利用这个相关模型计算 c_k 与 Δt_k。这里，我们将 τ 设为 3。在我们的经验中，这个取值能够较好地平衡准确性与效率。最后，我们利用上一步计算出来的 c_k 与 Δt_k 来计算增强词图，然后利用基于张量分解的方法（第 5.2.3节）来提取主题和主题之间的领先-滞后关系。在一台有着 Intel Xeon E52630 CPU（2.4 GHz）和 64GB 内存的机器上，处理数据集 A 的过程大概耗时 17 个小时，处理数据集 B 的过程耗时 9 个小时。表格 5.1总结了这两个数据集相关的统计信息。

表 5.1　两个数据集的统计信息总结。$|\mathcal{D}|$：推特消息的条数；$|\mathcal{U}|$：推特账号的个数；$|\mathcal{I}|$：主题个数；$|\mathcal{F}|$：存在领先-滞后关系的主题对个数；T：时间点个数。

	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{F}\rvert$	T
数据集 A	1,605,361	1,102	300	27,802	2,253
数据集 B	16,711,670	321,114	100	2,568	574

5.4.2　基准算法

我们将提出的算法与两个基准算法进行了比较。基准算法一是 Zhong 等人[79] 的算法。这个方法只用到了词的时间序列之间的协整关系来计算领先-滞后关系，而且它局限于两个文本源。为了把这个算法跟我们的算法进行比较，我们将这个算法中的张量替换为我们设计的张量，使得这个算法可以支持跟踪主题在多个文本源中的流动。基准算法二跟我们的算法相似。它跟我们算法的唯一不同是它不考虑词的时间序列之间的协整关系。我们设计基准算法二的目的是检查协整关系是否对于提高模型准确性有帮助。

5.4.3　实验设置

在这个实验中，我们衡量了模型的两种准确性：主题提取准确性以及领先-滞后提取准确性。这里，准确性等于标记为正确的主题（或者领先-滞后关系）比例[138]。我们邀请了两个专业为数据挖掘的博士生来标记主题或者领先-滞后关系是否是正确的。这两个博士生有数据标记方面的经验，并且对数据集较为熟悉。对于每个主题，我们给这两个博士生提供了主题中权重最高的 10 个关键词以及跟主题最相关，且彼此不是特别相似的 10 条推特消息。如果这个主题中的大部分关键词可以组成一个较为清楚易懂的想法或者观点，这个主题就会标记为正确的。我们之所以提供推特消息，是为了帮助这两个博士生更容易地理解这个主题。每个领先-滞后关系包含两个主题。对于每个领先-滞后关系，我们提供每个主题的权重最高的 10 个关键词以及一些推特消息对来解释这个领先-滞后关系。因为领先-滞后关系的数目特别多（表 5.1），我们只要这两个博士生标记了每个数据集的最重要（根据 $\mathbf{c}_k$ 计算）的 150 个领先-滞后关系。这两个博士生有 80.8% 的标记结果是一致的。为了进一步提高标记质量，我们聘请了一个专业编码员（Professional Coder）来检查标记结果并且纠正可能的错误。

5.4.4　实验结果

如表 5.2和表 5.3所示，我们的方法准确性远高于由 Zhong 等人[79] 提出的现有最先进的算法（基准算法一）。在这两个数据集上，我们的算法相比基准算法一，在主题准确性方面平均提高了 0.21，在领先-滞后关系准确性方面平均提高了 0.16。这个实验结果说明考虑推特消息内容以及元数据能够提高主题准确性。另外，我们的方法也比基准算法二的准确性更高。这说明考虑词序列的协整关系也能提高模型准确性。总的来说，综合考虑推特消息内容、元数据以及时间序列之间的协整关系可以提高模型准确性。

表 5.2　我们的方法和两个基准算法主题准确性对比。

	基准算法一	基准算法二	我们的算法
数据集 A	0.543	0.753	**0.793**
数据集 B	0.790	0.830	**0.940**

表 5.3　我们的方法和两个基准算法领先-滞后关系准确性对比。

	基准算法一	基准算法二	我们的算法
数据集 A	0.653	0.700	**0.800**
数据集 B	0.620	0.800	**0.803**

5.5　案例分析

5.5.1　美国议会

第一个案例分析是与一个政治传媒方面的教授（P1）共同进行的。这个教授希望理解不同政治党派以及重要政治人物在社交媒体上发言时采取的沟通策略，并且希望这个系统可以帮助他发现在各个社会议题上不同的政党以及政治人物扮演着什么样的角色。这个案例分析中，我们用到的数据集是数据集 A。

获取主题和主题流的概览（R1.1 和 R2.1）。为了理解推特上不同主题如何在政党成员被讨论，教授首先对第一层级的主题类进行了观察（图 5.1(a)）。他马上发现议会成员讨论的主题主要分为四类，这四类分别是经济相关（黄色）、总统与议会相关（蓝色）、节日等文化相关（绿色）以及 2016 年共和党总统候选人 Ted Cruz 相关（紫色）。因为对这些类别的主题如何在不同议会成员之间流动感兴趣，专家仔细检查了主题流视图（图 5.1(b)）。他发现不同主题类的流动规律差别很大。例如，Ted Cruz 相关的主题只在一个政党，即共和党中流动。与文化相关的主题（绿色）的强度随时间越来越不断增大，而且较为独立，跟其他主题类别之间的交互较少。教授还观察到经济相关主题（黄色）与总统、议会相关主题（蓝色）频繁地进行交互（例如在 A 处），这说明这两个主题类别之间有着相互影响。教授表示这是因为与政府财政支出等经济问题相关的法案会在议会进行辩论和投票。

探索感兴趣主题流的细节信息（R3.1 和 R3.2）。Ted Cruz 相关的主题因为只在共和党中流动而吸引了教授的注意。教授点击主题流，在信息面板中观察相应的细节（图 5.1(c)）。他发现，这个主题之所以出现，是因为 Ted Cruz 申请了很多账号，并且用这些账号发布了很多带有自己名字话题标签（Hashtag）的推特消息。Ted Cruz 之所以这么做，是为了在总统竞选之前进一步提高自己的声望（C）。但是从信息面板中可以看到，转发这类消息的主要是他自己的账号（D），说明他的宣传策略并不成功。

发现多层级文本源的领先-滞后关系（R2.2 和 R2.3）。教授对于研究共和党与民主党在不同主题上领先-滞后的规律比较感兴趣。因此他通过检查全局领先-滞后符号来研究哪些主题上民主党领先，哪些主题上共和党领先。一些例子如图 5.7所示。在检查了这些主题和它们的主题流以后，教授发现民主党和共和党比较容易在它们支

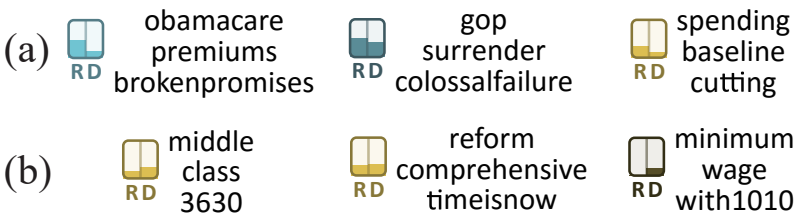


图 5.7　全局领先-滞后关系：(a) 共和党领先的主题；(b) 民主党领先的主题。

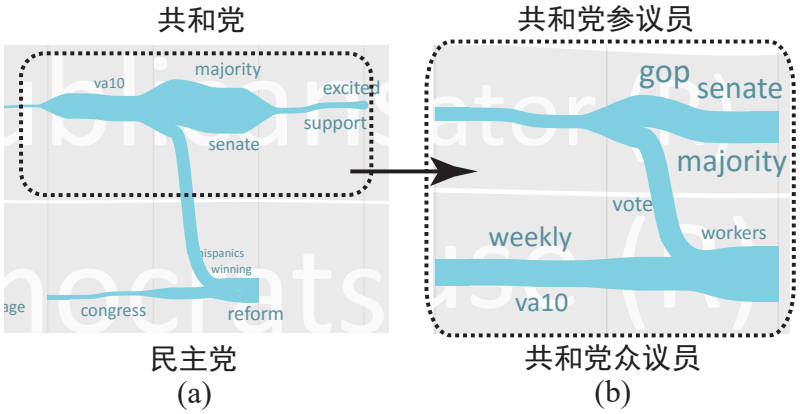


图 5.8　多层级文本源之间的领先-滞后关系：(a) 共和党对应文本源和民主党对应文本源之间的领先-滞后关系；(b) 共和党参议员和众议员对应文本源之间的领先-滞后关系。

持的主题上领先。以 GOP（Grand Old Party 的简称，共和党的别名）这个主题为例。图 5.8(a) 显示了这个主题在 GOP 选举党内领袖期间的流动情况。在那段时间，来自 GOP 的议会成员（例如 @SteveScalise、@RoyBlunt 以及 @SenPatRoberts）都积极发布关于共和党领袖选举的最新新闻。他们发布的推特消息的一个例子是“Early #FF for House #GOP leadership-elects: @GOPLeader @SteveScalise @cathymcmorris”。民主党在这个主题上落后是因为他们大多数是对共和党发的消息进行反击。他们发布的推特消息的一个例子是“Thought you'd see a different #GOP? Don't buy it...”。

教授对于在共和党中到底是参议院成员领先还是众议院成员领先感到比较好奇，因此他将共和党对应的文本源进行了分裂。如图 5.1(b) 所示，共和党对应的文本源分裂成了两个子文本源：共和党参议员对应的文本源和共和党众议员对应的文本源。教授通过观察图 5.1(b) 发现参议员在这个主题上领先于众议员。相关推特消息表明 2014 年 11 月 4 日，GOP 在参议院赢得多数选票。在这之后，共和党议员 Ron Johnson（@SenRonJohnson）在福克斯新闻中谈到了 GOP 下一步计划，并且发了相关的推特消息（“@SenRonJohnson will be on @foxandfriends at 6:15a CT to talk #GOP won majority in Senate and what's next. #BetterWithFriends @FoxNews”）。很多共和党的众议员紧跟着他进行了相关的讨论（“RT @HouseGOP: Watch @RepBradWenstrup deliver the Weekly GOP Address, discussing...”）。

放大研究感兴趣的主题类，并且对比多个时间点的主题流（R1.2 和 R2.2）。教授想知道为什么议会成员会讨论那么多跟政治无关的主题，因此他放大了节日等文化相关主题进行更仔细的研究。通过不断选择最大的子类进行放大，他放大到了最底层的主题类。他发现这个主题类中的五个主题都是与节日相关的，包括圣诞、母亲节以及复活节。教授从圣诞开始探索，希望理解为什么出现了这么多与节日相关的主题。图 5.1(a)、图 5.1(b) 以及图 5.1(c) 显示了圣诞相关主题在 2013 年、2014

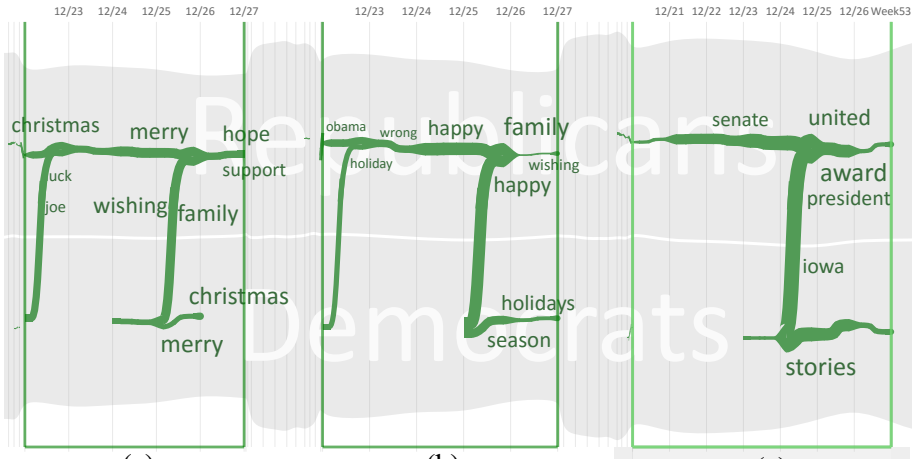


图 5.9　节日相关主题。

年以及 2015 年的主题流。教授发现圣诞周期间，在这个主题上始终都是共和党比民主党更加活跃。通过检查细节，教授发现共和党倾向于利用圣诞来达到自己的政治目的。例如，在 2014 年圣诞期间（图 5.1(b)），共和党提到了一些政治相关的词，例如“obama”、“wrong”。相关推特消息显示共和党这是在批评奥巴马（Obama）总统将军队派去了肯尼亚（Kenya），导致士兵无法回家过圣诞节（“Obama didn’t go to Kenya for Christmas. RT @RachelBynum: @HavanaTed @TeaPainUSA Going home for Christmas, what is wrong with that?”）。共和党利用这个机会来攻击支持奥巴马的民主党。这给支持总统奥巴马的民主党带来了公共关系危机。如图 5.1(b) 所示，两党派之间在 12 月 23 与 12 月 24 日没有什么交互，说明民主党在这件事情上并没有对共和党进行回应。相反地，民主党主要谈论的还是节日本身（关键词“holidays”）。随着圣诞节的到来，圣诞氛围越来越浓，最终推特上的讨论也趋于节日本身，民主党成功渡过此次公关危机。

教授进一步检查了其他的节日，例如独立日。经过探索，他发现共和党总是倾向于利用节日来达到自己的政治目的。他将这种现象称为节日的泛政治化。P1 评论说：“传媒研究人员和普通大众都有意识到泛政治化现象的存在[139]。但是，泛政治化的实证研究还很少。这个发现为美国议会中文化与节日相关主题的泛政治化提供了确凿的证据。”

5.5.2　埃博拉

埃博拉最严重的疫情是从 2013 年 12 月开始的。这个疫情直到 2016 年 3 月 16 日[140] 一共发现了 28,639 起病例，死亡人数达到 11,316 人。在这个案例分析中，我们与 P2 进行合作。她希望了解：1) 发生健康危机（如埃博拉）时，不同情感的主题怎么在社交网络中传播；2) 民众情感受病情严重性和舆情热度中哪个影响更大。

在了解上述两点以后，教授希望能够就如何采取合适的措施向政府和跨政府组织提出建议。这里用到的数据是数据集 B。

为了满足教授的分析需求，我们在系统中加入了如下三种信息：1) 主题流的情感；2) 情感强度与埃博拉病例数（埃博拉疫情严重性）的相关曲线；3) 情感强度与谈论埃博拉的推特消息条数（舆情热度）的相关曲线。实现时，我们利用基于词向量技术（Word Embedding）的情感分类方法[141] 来提取每条推特消息的情感。一个主题流在特定时间的情感通过将此时该主题流上所有推特消息的情感作平均得到。我们用从红色（消极情感）到灰色（中性情感）到绿色（积极情感）的颜色来展示情感。我们允许用户对这个颜色进行修改，以满足色弱、色盲人群的需求。我们利用皮尔森相关性来计算随时间变化的数值之间的相关程度。

比较疫情不同阶段的情感流动情况（R2.1 和 R2.3）。P2 对于受埃博拉严重影响的非洲国家非常关心。因此她搜索了“sierra leone”（塞拉利昂），即确诊埃博拉病例最多的非洲国家的名字，然后放大到了相应的主题类。这个主题类对应主题流如图 5.10所示。通过观察主题流上的情感和相应的关键词，教授发现塞拉利昂的疫情分为三个主要阶段。

第一个阶段是疫情爆发期（A）。在这个阶段，非洲国家和非非洲国家中都流动着强烈的负面情感。一些负面消息，例如罢工（D，“Sierra Leone burial teams on strike, leaving Ebola victims bodies in the streets.”）和死亡人数（E）吸引了公众的注意力。下方的线图显示在这个阶段，公众的情感与疫情严重性很相关，在相关度最高的时候皮尔森相关度达到了 0.72（F）。

第二个阶段是疫情衰减期（B）。在这个阶段，世界卫生组织 WHO 宣布塞拉利昂已经没有埃博拉病例了（“Ebola free”）。这时，强烈的正面情感开始在非非

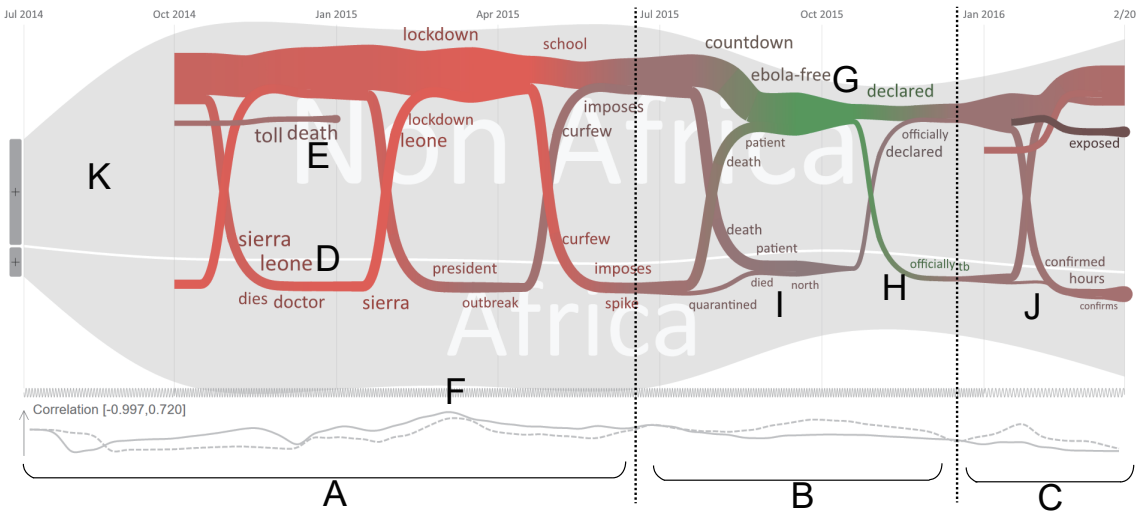


图 5.10　塞拉利昂相关主题上情感的流动情况。

洲国家中流动（G）。这个强烈的正面情感从某种程度上影响了非洲国家（H）。但是，非洲国家中仍然流动着负面情感（“Sierra Leone is Ebola free but a legacy of fear remains.”），而这种负面情感也影响了非非洲国家（I）。下方的相关曲线显示在这个阶段，比起疫情严重性，舆情热度对情感强烈程度的影响更大。

第三个阶段是疫情反弹期（C）。在这个阶段，一例新的病情在 WHO 宣布埃博拉的爆发已经结束以后几小时被发现（J，“Ebola resurfaces in Sierra Leone hours after WHO declares outbreak over http://cnn.it/233OoMT”）。如主题流所示，此时非非洲国家的负面情绪比非洲国家的更加严重。

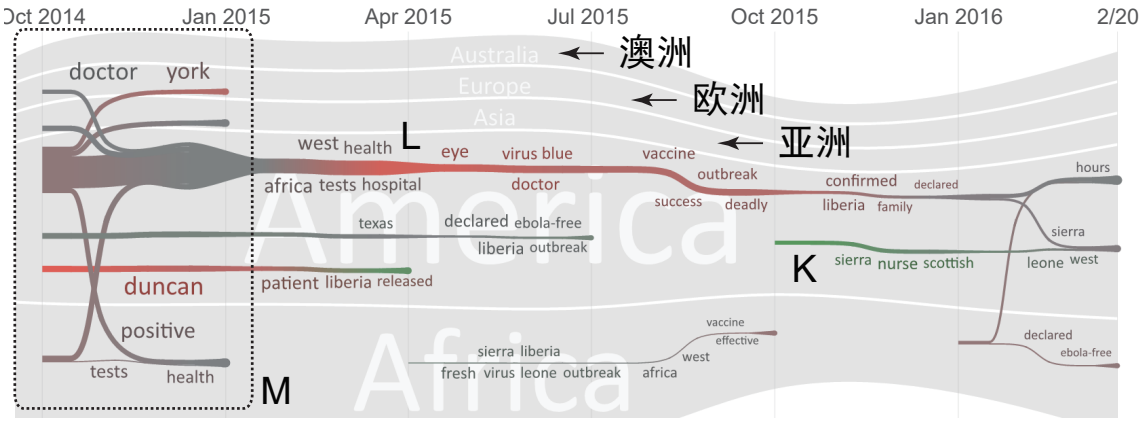
基于这些发现，P2 建议政府和跨政府组织在衰减期更加小心谨慎。她说尽管在这个阶段埃博拉病毒已经被很好地控制，但是由埃博拉带来的负面情绪仍然在社交媒体上流动，而这种负面情绪的“病毒”影响力并不比埃博拉病毒小。因此，在这个阶段引导民众更加理性地认识埃博拉非常关键。她接着建议例如 WHO 的国际组织避免过于绝对的言论，适当告知民众可能存在的风险，并且以直接、清晰的方式进行沟通。具体来说，她提出了下面的建议：

- 当疫情还在发展或者信息不够全面时，告知公众所发布消息的不确定性；
- 分享更多的信息而不是更少，否则民众可能认为有些重要的消息被隐藏了；
- 利用清晰、非技术的语言适当告知民众相应信息，例如用故事、具体的例子和趣闻轶事等来使得死板的数据变得生动活泼。

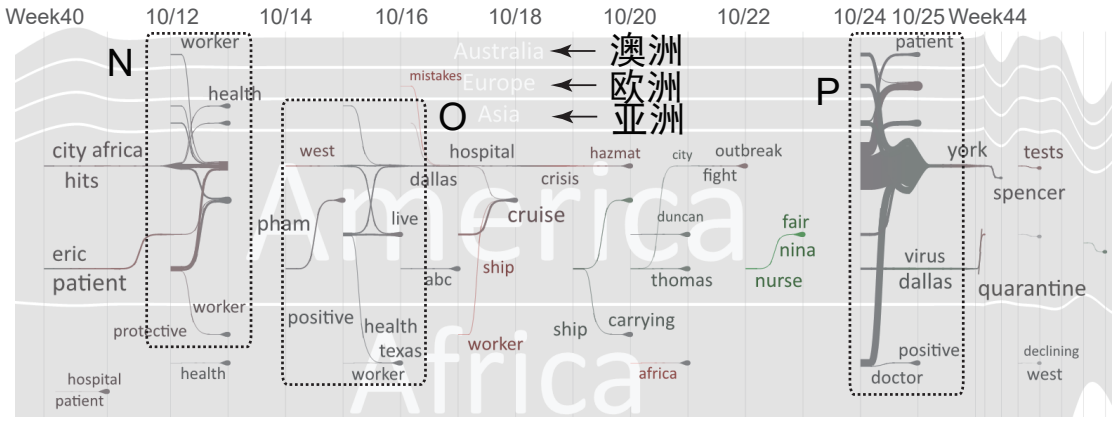
探索情感在多个大洲之间的流动（R2.1 和 R2.2）。在一些探索以后，P2 放大了一个与美国埃博拉病例相关的主题类，希望能够探索埃博拉在非洲以外的病例是如何对大众产生影响的。这个主题类的流如图 5.11(a) 所示。P2 将“non Africa”（非非洲）国家分裂成四个不同的大洲：“Australia”（澳大利亚）、“Europe”（欧洲）、“Asia”（亚洲）以及“America”（美洲），并且分析主题流是如何在不同大洲间流动的。通过分析图 5.11(a) 中的主题流，她发现了如下有趣的现象：

- 澳大利亚是这些洲里面被影响程度最小的；
- 美洲的强烈正面情感（K）和强烈负面情感（L）对其他洲的影响都比较小；
- 很多洲在 2014 年 10 月到 2015 年 1 月这段时间都受到了较大的影响（M）。在这段时间里，埃博拉在美国的前几个病例被发现了（“Texas nurse tests positive for Ebola, would be 1st Ebola transmission in U.S.”）。

P2 对于美国的前几例埃博拉病例产生的影响很感兴趣，因此她放大到了相应时间片段（M）来检查更多的细节。在迭代式地扩大有着最多的主题流的时间片段以后，她放大到了美国前三例医护人员案例发现的时间段（2014 年 10 月 11 日到 2014 年 10 月 25 日）。图 5.11(b) 展示了这段时间主题在多个大洲的流动情况。



(a)



(b)

图 5.11　主题流在多个大洲之间的流动情况。

这三个病例的主题流分别标记为 N、O 和 P。P2 发现第三例案例是最具有影响力的。她对这个发现比较惊讶。在她设想当中，第一例的影响力应该是最大的。为了解释这个现象，她在推特消息中找到这三个案例的病人信息（Who）、位置信息（Where）以及日期（When），然后就这三个方面对这三个病例进行了比较。在进行比较后，她发现第三个病例与前两个病例的主要区别在病人上。前两个病例中的病人都是来自德克萨斯州医院的护士，而第三个病例的病人是来自无国界组织（Doctors without Borders）的医生。无国界组织是一个著名的国际非政府组织。P2 评论到因为无国界组织在公共健康议题上非常活跃，这个组织中的医生患病比其他人患病受到更多的关注是非常合理的。

5.6　局限性讨论

尽管数值实验和案例分析展示了我们系统的有效性，我们的系统仍然有一些局限之处。

首先，我们提取主题及其领先-滞后关系的算法是离线算法。这个算法的主要计算代价来源于时间复杂度较高的张量分解。在进行张量分解时，最高的代价是计算每个维度的协方差矩阵[142]。因此，总体的时间复杂度是 $O(N^{M+1})$。这里 M 是张量的维度，N 是每个维度上的平均元素数。这个算法在有百万条推特消息的数据集上往往需要几个小时的时间。因此，它并不能支持实时的参数调整，也不能在新的推特消息到来时实时计算领先-滞后关系。一个可能的解决方案是利用在线张量分解算法来增量式地对主题以及主题领先-滞后关系进行更新。

第二个局限性是模型中的主题个数需要手动进行设定，这限制了我们的方法对于数据集的可移植性。这个问题可以通过用赤池信息准则（AIC，Akaike Information Criterion）[143] 自动选择主题数目来解决。

第三个局限性是可以清晰展示的主题类个数有限。这是因为我们利用颜色来帮助用户找到气泡树上主题类和流视图中主题流的对应关系。我们通过利用关键词和交互来解决这个问题。具体来说，用户可以通过 1) 阅读主题类以及主题流旁的关键词，2) 将鼠标悬停在主题类上来对主题流进行高亮（反之亦然）来找到主题类和主题流之间的对应关系。

5.7　小结及结论

在这篇文章中，我们设计了一个交互式可视分析系统来帮助用户探索相关主题在不同文本源之间的流动。我们的系统能够自动挖掘一组不相同但是相关的来自于不同文本源的主题之间的领先-滞后关系。另外，我们还设计了一个包含气泡树、流向图和焦点加上下文的时间轴的可视化来从不同层级展示挖掘结果。数值实验和两个与领域专家合作进行的案例分析展示了我们方法的有效性和有用性。

专家的反馈意见为我们的未来工作提供了方向。例如，专家们希望可以在主题流上进行假设检验。具体来说，他们希望可以修改文本源上特定时间点主题之间的领先-滞后关系，然后看到整体领先-滞后关系如何变化。为了满足这个需求，我们希望研究一些可能的支持假设检验的算法。除此之外，我们还希望能够比较不同的对主题流布局算法产生的歧义（例如点重叠、边交叉、边交叉角度以及点边重叠），从而定量验证可视化布局算法的有效性。

第 6 章　总结与展望

6.1　本文工作总结

本论文提出了复杂文本的主题挖掘及可视分析方法，帮助用户快速、有效地分析单源动态文本、多源静态文本、多源动态文本中的大量丰富的主题信息（图 6.1）。具体来说，我们的主要创新点如下。

单源动态文本方面，本论文提出了一种快速、有效地对大量主题的分裂、合并关系进行分析的数据挖掘及可视分析方法。该方法利用多分枝主题树组织主题，通过分析主题树随时间的动态变化研究大量主题的分裂、合并关系。为了准确地挖掘动态多分枝主题树，本论文设计了一个贝叶斯在线滤波框架。这个框架可以同时优化拟合度和平滑度。为了解决现有方法产生结果不够平滑的问题，我们引入了三元组约束与扇形约束。为了提高算法的效率，我们建立了约束树，将算法时间复杂度从 $\Omega(n^3)$ 降低到了 $O(n\log(n))$。最后，我们利用 TextFlow[1] 对主题挖掘结果进行可视化，使得结果直观易于理解。

多源静态文本方面，本论文提出了对多个文本源中大量共有与独有主题进行挖掘和可视分析的方法。我们首先提取每个文本源的主题图。然后我们利用设计的一致的图匹配算法将这些主题图进行匹配。这个一致的图匹配算法的特点是在

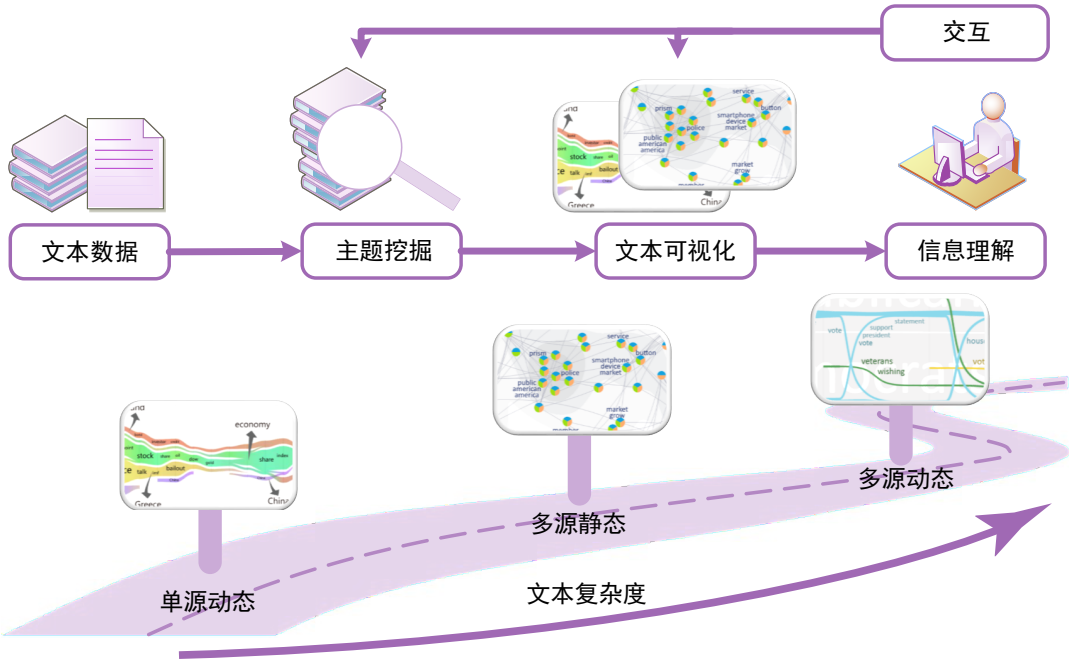


图 6.1　本论文提出了单源动态文本、多源静态文本以及多源动态文本的主题挖掘及可视分析方法，帮助用户快速、有效地分析复杂文本中大量主题的丰富信息。

对两张以上的主题图进行匹配时，也能保证匹配结果是一致的（不互相矛盾的）。接着，我们允许用户根据自己的信息需求对匹配结果进行修改。为了有效地利用用户反馈生成符合用户需求的主题全景图，我们提出基于度量学习与特征选择的增量式图匹配结果修改算法。最后，为了有效展示大量的共有主题与独有主题，我们设计了基于 LOD 的可视化方法。该方法结合径向冰柱树与基于密度的图可视化有效展示大量主题，同时保证用户可以自由地对主题全景图进行放大与缩小。这个可视化利用基于 Voronoi 剖分的布局算法有效区分共有主题和独有主题，保证全景图的可读性。

多源动态文本方面，本论文提出了对多个文本源中大量相关主题的领先-滞后关系进行挖掘和可视分析的方法。为了准确提取主题和它们之间的领先-滞后关系，我们开发了基于随机游走的挖掘模型。该模型利用随机游走相关模型综合考虑文本内容、词时间序列的协整关系以及文本元数据（如转发关系等），提高了多源动态主题提取与领先-滞后关系提取的准确性。另外，该模型利用张量统一考虑多个文本源，可以计算三个及以上文本源相关主题之间的领先-滞后关系。为了减少展示大量领先-滞后关系产生的视觉混乱和歧义，我们设计了基于 Voronoi 树图的气泡树、基于相关聚类的流向图以及焦点加上下文的时间轴，使得用户可以快速有效地分析涉及多个文本源、多个主题、多个时间点的领先-滞后关系。

6.2　未来工作展望

复杂文本中的主题挖掘与可视分析具有较为广泛的研究前景。下面，我们从数据、算法、结果三个方面对未来可能进行的研究进行简单的探讨。

数据方面，我们在考虑将文本数据与一些重要的时间序列，例如股票数据、民众支持率等，综合在一起进行研究。之所以考虑这么做，是因为进行案例分析时，专家们经常提到希望综合文本数据来分析他们平时关注的这些时间序列。

算法方面，我们希望将现有的算法（如提取主题领先-滞后关系的算法）进行拓展，使得它们能够在线地对文本流进行分析。现实生活中，人们对于在线算法有较大的需求。这是因为人们关注的事件往往是正在进行中的。因此，事件相关的文档不会在第一次分析时全部得到，而是会随着时间不断地到来。只有设计出适合文本流的在线算法，才能帮助用户快速发现值得注意的信息。

结果方面，我们希望能提取更为丰富的主题信息，例如主题之间的因果关系。我们原来的算法只能计算两个主题是否相关，不能判断一个主题是否导致了另一个主题的发生。因果关系分析是一个很有用但是非常难的问题。我们考虑的是通过可视化技术，将人的知识经验引入到因果关系提取中，从而提高算法准确性。

参考文献

[1] Cui W, Liu S, Tan L, et al. Textflow: Towards better understanding of evolving topics in text. IEEE TVCG, 2011, 17(12):2412–2421.

[2] Wang X, Liu S, Song Y, et al. Mining evolutionary multi-branch trees from text streams. KDD, 2013. 722–730.

[3] Liu S, Chen Y, Wei H, et al. Exploring topical lead-lag across corpora. IEEE TKDE, 2015, 27(1):115–129.

[4] Dou W, Yu L, Wang X, et al. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. IEEE TVCG, 2013, 19(12):2002–2011.

[5] Blundell C, Teh Y W, Heller K A. Bayesian rose trees. UAI, 2010. 65–72.

[6] Liu S, Wang X, Chen J, et al. Topicpanorama: a full picture of relevant topics. IEEE VAST, 2014. 183–192.

[7] Twitter. https://twitter.com/, 2016.

[8] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. JMLR, 2003, 3:993–1022.

[9] Blei D M, Lafferty J D. Dynamic topic models. ICML, 2006. 113–120.

[10] Blei D, Lafferty J. Correlated topic models. NIPS. 2006: 147–154.

[11] Wise J A, Thomas J J, Pennock K, et al. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. IEEE InfoVis, 1995. 51–58.

[12] Havre S, Hetzler E G, Whitney P, et al. Themeriver: visualizing thematic changes in large document collections. IEEE TVCG, 2002, 8(1):9–20.

[13] Gao Z J, Song Y Q, Liu S X, et al. Tracking and connecting topics via incremental hierarchical dirichlet processes. ICDM, 2011. 1056–1061.

[14] Ahmed A, Ho Q, Eisenstein J, et al. Unified analysis of streaming news. WWW, 2011. 267–276.

[15] Wang X, Zhai C, Roth D. Understanding evolution of research themes: a probabilistic generative model for citations. KDD, 2013. 1115–1123.

[16] Liu S, Zhou M X, Pan S, et al. Interactive, topic-based visual text summarization and analysis. CIKM, 2009. 543–552.

[17] Liu S, Zhou M X, Pan S, et al. TIARA: Interactive, topic-based visual text summarization and analysis. ACM TIST, 2012, 3(2):25:1–25:28.

[18] Pan S, Zhou M X, Song Y, et al. Optimizing temporal topic segmentation for intelligent text visualization. IUI, 2013. 339–350.

[19] Dörk M, Gruen D M, Williamson C, et al. A visual backchannel for large-scale events. IEEE TVCG, 2010, 16(6):1129–1138.

[20] Dou W, Wang X, Chang R, et al. Paralleltopics: A probabilistic approach to exploring document collections. IEEE VAST, 2011. 231–240.

[21] Gad S, Javed W, Ghani S, et al. Themedelta: Dynamic segmentations over temporal topic models. IEEE TVCG, 2015, 21(5):672–685.

[22] Xu P, Wu Y, Wei E, et al. Visual analysis of topic competition on social media. IEEE TVCG, 2013, 19(12):2012–2021.

[23] Sun G, Wu Y, Liu S, et al. EvoRiver: Visual analysis of topic coopetition on social media. IEEE TVCG, 2014..

[24] Wagstaff K, Cardie C. Clustering with instance-level constraints. ICML, 2000. 1103–1110.

[25] Basu S, Banerjee A, Mooney R J. Semi-supervised clustering by seeding. ICML, 2002. 27–34.

[26] Miyamoto S, Terami A. Constrained agglomerative hierarchical clustering algorithms with penalties. IEEE FUZZ, 2011. 422–427.

[27] Schultz M, Joachims T. Learning a distance metric from relative comparisons. NIPS, 2003. 41–48.

[28] Kumar N, Kummamuru K, Paranjpe D. Semi-supervised clustering with metric learning using relative comparisons. ICDM, 2005. 693–696.

[29] Bade K, N$^{'}$urnberger A. Creating a cluster hierarchy under constraints of a partially known hierarchy. SDM, 2008. 13–24.

[30] Zheng L, Li T. Semi-supervised hierarchical clustering. ICDM, 2011. 982–991.

[31] Zhao H F, Qi Z J. Hierarchical agglomerative clustering with ordering constraints. WKDD, 2010. 195–199.

[32] Liu E Y, Zhang Z J, Wang W. Clustering with relative constraints. KDD, 2011. 947–955.

[33] Zhao J, Cao N, Wen Z, et al. # fluxflow: Visual analysis of anomalous information spreading on social media. IEEE TVCG, 2014, 20(12):1773–1782.

[34] Teh Y W, Jordan M I, Beal M J, et al. Sharing clusters among related groups: Hierarchical Dirichlet processes. NIPS, volume 17, 2005.

[35] Teh Y W, Jordan M I, Beal M J, et al. Hierarchical dirichlet processes. Journal of the American Statistical Association, 2004, 101.

[36] Zhai C, Velivelli A, Yu B. A cross-collection mixture model for comparative text mining. KDD, 2004. 743–748.

[37] Paul M, Girju R. Cross-cultural analysis of blogs and forums with mixed-collection topic models. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2009. 1408–1417.

[38] Wang C, Thiesson B, Meek C, et al. Markov topic models. AISTATS, 2009. 583–590.

[39] Shen Z Y, Sun J, Shen Y D. Collective latent dirichlet allocation. ICDM, 2008. 1019–1024.

[40] Cao N, Gotz D, Sun J, et al. Solarmap: Multifaceted visual analytics for topic exploration. ICDM, 2011. 101–110.

[41] Oelke D, Strobelt H, Rohrdantz C, et al. Comparative exploration of document collections: A visual analytics approach. Computer Graphics Forum, 2014, 33(3):201–210.

[42] Gleicher M, Albers D, Walker R, et al. Visual comparison for information visualization. Information Visualization, 2011, 10(4):289–309.

[43] Landesberger T, Kuijper A, Schreck T, et al. Visual analysis of large graphs: State-of-the-art and future research challenges. Computer Graphics Forum, 2011, 30(6):1719–1749.

[44] Alper B, Bach B, Riche N H, et al. Weighted graph comparison techniques for brain connectivity analysis. CHI, 2013. 483–492.

[45] Branke J. Dynamic graph drawing. Drawing Graphs, 1999. 228–246.

[46] Cui W, Wang X, Liu S, et al. Let it flow: a static method for exploring dynamic graphs. IEEE PacificVis, 2014. 121–128.

[47] Kumar G, Garland M. Visual exploration of complex time-varying graphs. IEEE TVCG, 2006, 12(5):805–812.

[48] Andrews K, Wohlfahrt M, Wurzinger G. Visual graph comparison. IEEE InfoVis, 2009. 62–67.

[49] Munzner T, Guimbretière F, Tasiran S, et al. Treejuxtaposer: scalable tree comparison using focus+ context with guaranteed visibility. ACM TOG, volume 22, 2003. 453–462.

[50] Collins C, Carpendale M S T. Vislink: Revealing relationships amongst visualizations. IEEE TVCG, 2007, 13(6):1192–1199.

[51] Bremm S, Landesberger T, Hess M, et al. Interactive visual comparison of multiple trees. IEEE VAST, 2011. 31–40.

[52] Vehlow C, Reinhardt T, Weiskopf D. Visualizing fuzzy overlapping communities in networks. IEEE TVCG, 2013, 19(12):2486–2495.

[53] Conte D, Foggia P, Sansone C, et al. Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18(03):265–298.

[54] Riesen K, Bunke H. Approximate graph edit distance computation by means of bipartite graph matching. Image Vision Computing, 2009, 27(7):950–959.

[55] Riesen K, Jiang X, Bunke H. Exact and inexact graph matching: Methodology and applications. Managing and Mining Graph Data. 2010: 217–247.

[56] Gao X, Xiao B, Tao D, et al. A survey of graph edit distance. Pattern Analysis and Applications, 2010, 13(1):113–129.

[57] Suganthan P N, Teoh E K, Mital D P. Pattern recognition by graph matching using the potts mft neural networks. Pattern Recognition, 1995, 28(7):997–1009.

[58] Wilson R C, Hancock E R. Structural matching by discrete relaxation. IEEE PAMI, 1997, 19(6):634–648.

[59] Wilson R C, Hancock E R, Luo B. Pattern vectors from algebraic graph theory. IEEE PAMI, 2005, 27(7):1112–1124.

[60] Neuhaus M, Bunke H. A convolution edit kernel for error-tolerant graph matching. ICPR, volume 4, 2006. 220–223.

[61] Myers R, Wison R, Hancock E R. Bayesian graph edit distance. IEEE PAMI, 2000, 22(6):628–635.

[62] Yan J, Tian Y, Zha H, et al. Joint optimization for consistent multiple graph matching. ICCV, 2013. 1649–1656.

[63] Williams M L, Wilson R C, Hancock E R. Multiple graph matching with bayesian inference. Pattern Recognition Letters, 1997, 18(11-13):1275–128.

[64] Solé-Ribalta A, Serratosa F. On the computation of the common labelling of a set of attributed graphs. CIARP, 2009. 137–144.

[65] Ribalta A, Serratosa F. Models and algorithms for computing the common labelling of a set of attributed graphs. Computer Vision and Image Understanding, 2011, 115(7):929–945.

[66] Sambasivan R R, Shafer I, Mazurek M L, et al. Visualizing request-flow comparison to aid performance diagnosis in distributed systems. IEEE TVCG, 2013, 19(12):2466–2475.

[67] Hascoët M, Dragicevic P. Interactive graph matching and visual comparison of graphs and clustered graphs. AVI, 2012. 522–529.

[68] Wang X, Zhang K, Jin X, et al. Mining common topics from multiple asynchronous text streams. WSDM, 2009. 192–201.

[69] Zhang J, Song Y, Zhang C, et al. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. KDD, 2010. 1079–1088.

[70] Hong L, Dom B, Gurumurthy S, et al. A time-dependent topic model for multiple text streams. KDD, 2011. 832–840.

[71] Lloyd L, Kaulgud P, Skiena S. Newspapers vs. blogs: Who gets the scoop? AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006. 117–124.

[72] Shi X, Nallapati R, Leskovec J, et al. Who leads whom: Topical lead-lag analysis across corpora. NIPS Workshop, 2010.

[73] Leskovec J, Backstrom L, Kleinberg J. Meme-tracking and the dynamics of the news cycle. KDD, 2009. 497–506.

[74] Wu F, Song Y, Liu S, et al. Lead-lag analysis via sparse co-projection in correlated text streams. CIKM, 2013. 2069–2078.

[75] Gerrish S, Blei D M. A language-based approach to measuring scholarly impact. ICML, volume 10, 2010. 375–382.

[76] Nallapati R, Mcfarland D A, Manning C D. Topicflow model: Unsupervised learning of topic-specific influences of hyperlinked documents. AISTATS, 2011. 543–551.

[77] Shaparenko B, Joachims T. Information genealogy: Uncovering the flow of ideas in non-hyperlinked document databases. KDD, 2007. 619–628.

[78] Nallapati R, Shi X, McFarland D A, et al. Leadlag lda: Estimating topic specific leads and lags of information outlets. ICWSM, 2011. 558–561.

[79] Zhong Y, Liu S, Wang X, et al. Tracking idea flows between social groups. To appear in AAAI, 2016.

[80] Dou W, Wang X, Skau D, et al. Leadline: Interactive visual analysis of text data through event identification and exploration. IEEE VAST, 2012. 93–102.

[81] Luo D, Yang J, Krstajic M, et al. Eventriver: Visually exploring text collections with temporal references. IEEE TVCG, 2012, 18(1):93–105.

[82] Cui W, Liu S, Wu Z, et al. How hierarchical topics evolve in large text corpora. To appear in IEEE TVCG, 2014..

[83] Liu S, Yin J, Wang X, et al. Online visual analytics of text streams. To appear in IEEE TVCG, 2015. 1–15.

[84] Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering. KDD, 2006. 554–560.

[85] Bing news. http://www.bing.com/news, March, 2014.

[86] Liu X, Song Y, Liu S, et al. Automatic taxonomy construction from keywords. KDD, 2012. 1433–1441.

[87] Madsen R E, Kauchak D, Elkan C. Modeling word burstiness using the Dirichlet distribution. ICML, 2005. 545–552.

[88] Basu S, Bilenko M, Mooney R J. A probabilistic framework for semi-supervised clustering. KDD, 2004. 59–68.

[89] Ng M P, Wormald N C. Reconstruction of rooted trees from subtrees. Discrete Applied Mathematics, 1996, 69(1-2):19–31.

[90] Blundell C, Teh Y W, Heller K. Discovering non-binary hierarchical structures with Bayesian rose trees. Mixture Estimation and Applications. 2011: 161–187.

[91] 20 newsgroups dataset. http://qwone.com/~jason/20Newsgroups/.

[92] Heller K A, Ghahramani Z. Bayesian hierarchical clustering. ICML, 2005. 297–304.

[93] Strehl A, Ghosh J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. JMLR, 2003, 3:583–617.

[94] 20 newsgroups dataset. http://nytimes.com.

[95] Robinson D F, Foulds L R. Comparison of phylogenetic trees. Mathematical Biosciences, 1981, 53:131–147.

[96] Lin Y, Rajan V, Moret B M E. A metric for phylogenetic trees based on matching. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2012, 9(4):1014–1022.

[97] Chen J, Zhu J, Wang Z, et al. Scalable inference for logistic-normal topic models. NIPS. 2013: 2445–2453.

[98] Salomatin K, Yang Y, Lad A. Multi-field correlated topic modeling. SDM, 2009. 628–637.

[99] Cortés X, Serratosa F, Solé-Ribalta A. Active graph matching based on pairwise probabilities between nodes. Structural, Syntactic, and Statistical Pattern Recognition. 2012: 98–106.

[100] Yantis S. Multielement visual tracking: Attention and perceptual organization. Cognitive Psychology, 1992, 24(3):295–340.

[101] Korsah G A, Stentz A T, Dias M B. The dynamic hungarian algorithm for the assignment problem with changing costs. Technical Report CMU-RI-TR-07-27, July, 2007.

[102] Jain P, Kulis B, Dhillon I S, et al. Online metric learning and fast similarity search. NIPS, 2009. 761–768.

[103] Pocock A C. Feature selection via joint likelihood[D]. University of Manchester, 2012.

[104] Guyon I, Elisseeff A. An introduction to variable and feature selection. JMLR, 2003, 3:1157–1182.

[105] McLachlan P, Munzner T, Koutsofios E, et al. Liverac: interactive visual exploration of system management time-series data. CHI, 2008. 1483–1492.

[106] Wu Y, Wei F, Liu S, et al. Opinionseer: Interactive visualization of hotel customer feedback. IEEE TVCG, 2010, 16(6):1109–1118.

[107] Wu Y, Yuan G X, Ma K L. Visualizing flow of uncertainty through analytical processes. IEEE TVCG, 2012, 18(12):2526–2535.

[108] MacEachren A M, Roth R E, O'Brien J, et al. Visual semiotics & uncertainty visualization: An empirical study. IEEE TVCG, 2012, 18(12):2496–2505.

[109] Kamada T, Kawai S. An algorithm for drawing general undirected graphs. Information Processing Letters, 1989, 31(1):7–15.

[110] Misue K, Eades P, Lai W, et al. Layout adjustment and the mental map. Journal of Visual Languages and Computing, 1995, 6(2):183–210.

[111] Balzer M, Deussen O. Voronoi treemaps. IEEE InfoVis, 2005. 49–56.

[112] Lampe O D, Hauser H. Interactive visualization of streaming data with kernel density estimation. IEEE PacificVis, 2011. 171–178.

[113] Bach B, Pietriga E, Fekete J D. Graphdiaries: animated transitions andtemporal navigation for dynamic networks. IEEE TVCG, 2014, 20(5):740–754.

[114] Willett W, Heer J, Agrawala M. Scented widgets: Improving navigation cues with embedded visualizations. IEEE TVCG, 2007, 13(6):1129–1136.

[115] Boardreader. http://www.boardreader.com, March, 2014.

[116] Choo J, Lee C, Reddy C K, et al. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. IEEE TVCG, 2013, 19(12):1992–2001.

[117] Liu S, Chen Y, Wei H, et al. Exploring topical lead-lag across corpora. IEEE TKDE, 2015, 27(1):115–129.

[118] Bracegirdle C, Barber D. Bayesian conditional cointegration. ICML, 2012. 1095–1102.

[119] Kolda T, Bader B, Kenny J. Higher-order web link analysis using multilinear algebra. ICDM, 2005. 242–249.

[120] Munzner T. A nested process model for visualization design and validation. IEEE TVCG, 2009, 15(6):921–928.

[121] Van Zoonen L. Feminist media studies, volume 9. Sage, 1994.

[122] Liu M, Liu S, Zhu X, et al. An uncertainty-aware approach for exploratory microblog retrieval. IEEE TVCG, 2016, 22(1):250–259.

[123] Shahaf D, Guestrin C. Connecting the dots between news articles. KDD, 2010. 623–632.

[124] Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. ACM SIGIR, 2003. 267–273.

[125] Ware C. Information visualization: perception for design. Elsevier, 2012.

[126] Beck F, Koch S, Weiskopf D. Visual analysis and dissemination of scientific literature collections with survis. IEEE TVCG, 2016, 22(1):180–189.

[127] Simonetto P, Archambault D, Scheideg C. A simple approach for boundary improvement of euler diagrams. IEEE TVCG, 2016, 22(1):678–687.

[128] Bach B, Pietriga E, Fekete J D. Visualizing dynamic networks with matrix cubes. CHI, 2014. 877–886.

[129] Vatti B R. A generic solution to polygon clipping. Communications of the ACM, 1992, 35(7):56–63.

[130] Wu Y, Liu S, Yan K, et al. Opinionflow: Visual analysis of opinion diffusion on social media. IEEE TVCG, 2014, 20(12):1763–1772.

[131] Riehmann P, Hanfler M, Froehlich B. Interactive sankey diagrams. IEEE InfoVis, 2005. 233–240.

[132] Cui W, Zhou H, Qu H, et al. Geometry-based edge clustering for graph visualization. IEEE TVCG, 2008, 14(6):1277–1284.

[133] Buchin K, Speckmann B, Verbeek K. Flow map layout via spiral trees. IEEE TVCG, 2011, 17(12):2536–2544.

[134] Phan D, Xiao L, Yeh R, et al. Flow map layout. IEEE InfoVis, 2005. 219–224.

[135] Wang Y, Shen Q, Archambault D, et al. Ambiguityvis: Visualization of ambiguity in graph layouts. IEEE TVCG, 2016, 22(1):359–368.

[136] Kirk P, Griffin J E, Savage R S, et al. Bayesian correlated clustering to integrate multiple datasets. Bioinformatics, 2012, 28(24):3290–3297.

[137] Tweetinvi: a .net library that provides an access to Twitter APIs. https://tweetinvi.codeplex.com/, March, 2016.

[138] The definition of accuracy in Wikipedia. https://en.wikipedia.org/wiki/Accuracy_and_precision, March, 2016.

[139] McCright A M, Dunlap R E. The politicization of climate change and polarization in the american public's views of global warming, 2001–2010. The Sociological Quarterly, 2011, 52(2):155–194.

[140] Ebola situation report. http://apps.who.int/ebola/current-situation/ebola-situation-report-16-march-2016, March, 2016.

[141] Tang D, Wei F, Yang N, et al. Learning sentiment-specific word embedding for Twitter sentiment classification. ACL, 2014. 1555–1565.

[142] Sun J, Tao D, Papadimitriou S, et al. Incremental tensor analysis: Theory and applications. TKDD, 2008, 2(3):11:1–11:37.

[143] Akaike H. Akaike's Information Criterion. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011: 25–25.

致　谢

五年多的博士生活是我人生中最重要的一段时光。这段时间里，我遇到了对我影响很大的老师和同学们。你们的智慧、严谨、温柔和热情让我在博士这条原本孤独的道路上可以不断前行。你们的师恩和友情我会一直铭记。

感谢导师郭百宁教授和温江涛教授。在你们的信任和鼓励下，我找到了自己喜欢的科研方向。也是你们的支持让我走出最迷茫的时期，取得一定科研成果。你们的教诲将激励我在今后的科研道路上严谨勤奋、不断创新。

特别感谢亲自教会我科研的刘世霞教授。在跟您学习的这五年多时间里，您从科研方法、研究态度、为人处世各个方面言传身教，让我可以成为一个合格的博士生，也成为一个更好的人。没有您的教导和温柔关怀，我不可能如此顺利地完成博士学业。未来的日子不能时刻留在您身边，但还是希望能不断听到您的教诲。

感谢高等研究中心和微软亚洲研究院给我提供理想的学习机会与学习环境。谢谢吴念乐老师、李丽老师。是你们用辛勤、细致的工作帮我们处理学校各项事务，为我们营造出良好的生活和学习环境。谢谢微软亚洲研究院的研究员，特别是童欣研究员、宋阳秋研究员和崔为炜研究员，你们的真知灼见给了我不少的帮助。

感谢我所有的科研合作者们，尤其是密歇根州立大学的彭文森教授、清华大学的苏晋教授以及清华大学的朱军教授。你们给了我莫大的帮助，让我可以顺利完成各个科研项目。

感谢一直陪伴我的师弟师妹以及朋友们，你们的机敏聪明、勤奋踏实总是给我惊喜，你们的无私帮助也总让我感动。尤其感谢刘梦尘师弟，在跟你一起学习的三年多的时间里，我从你身上学到了很多，你的善解人意和帮助不时给我温暖。谢谢我亲爱的室友们、微软亚洲研究院一起实习的朋友们和大学、高中的朋友们一直以来的友谊，你们的陪伴帮助我度过困难的时刻。

最后，谨以此文献给我的爸爸妈妈。一直以来有你们的爱是我最大的幸运。你们事事以我为优先，我却不能经常在你们身边陪伴。希望你们能够一直身体健康、平安幸福。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，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，本学位论文的研究成果不包含任何他人享有著作权的内容。对本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。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1989 年 08 月 21 日出生于湖南省桃江县。

2007 年 9 月考入清华大学电子工程系电子信息科学与技术专业，2011 年 7 月本科毕业并获得工学学士学位。

2011 年 9 月免试进入清华大学高等研究院攻读工学博士学位至今。

发表的学术论文

[1] Wang X T, Liu S X, Song Y Q, Guo B N. Mining Evolutionary Multi-Branch Trees from Text Streams. ACM SIGKDD Conferences on Knowledge Discovery and Data Mining, 2013, 722-730.

[2] Wang X T, Liu S X, Liu J L, Chen J F, Zhu J, Guo B N. TopicPanorama: A Full Picture of Relevant Topics. In press. (已被 IEEE Transactions on Visualization and Computer Graphics 录用).

[3] Wang X T, Liu S X, Chen Y, Peng T-Q, Su J, Yang J, and Guo B N. How Ideas Flow across Multiple Social Groups. In press. (已被 IEEE Conference on Visual Analytics Science and Technology 录用).

[4] Liu S X, Wang X T, Song Y Q, Guo B N. Evolutionary Bayesian Rose Trees. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(6): 1533-1546.

[5] Liu S X, Wang X T, Chen J F, Zhu J, Guo B N. TopicPanorama: a Full Picture of Relevant Topics. Proceedings of IEEE Conference on Visual Analytics Science and Technology, 2014, 183–192.

[6] Cui W W, Wang X T, Liu S X, Riche N H, Madhyastha T M, Ma K-L, Guo B N. Let It Flow: a Static Method for Exploring Dynamic Graphs. Proceedings of IEEE Pacific Visualization Symposium, 2014, 121 - 128.

[7] Liu S X, Yin J L, Wang X T, Cui W W, Cao K L, Pei J. Online Visual Analytics of Text Streams. In press. (已被 IEEE Transactions on Visualization and Computer Graphics 录用).

[8] Zhong Y X, Liu S X, Wang X T, Xiao J N, Song Y Q. Tracking Idea Flows between Social Groups. In press. (已被 AAAI Conference on Artificial Intelligence 录用).