

密级： 保密期限：

北京郵電大學

硕士学位论文



题目：自适应主题模型及其在推荐系统
中的应用

学 号：2014110022

姓 名：周国睿

专 业：信息与通信工程

导 师：陈光

学 院：信息与通信工程学院

二〇一六年十二月二十五日

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构递交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

自适应主题模型及其在推荐系统中的应用

摘要

中、英文摘要位于声明的次页，摘要应简明表达学位论文的内容要点，体现研究工作的核心思想。重点说明本项科研的目的和意义、研究方法、研究成果、结论，注意突出具有创新性的成果和新见解的部分。

关键词是为文献标引工作而从论文中选取出来的、用以表示全文主题内容信息的术语。关键词排列在摘要内容的左下方，具体关键词之间以均匀间隔分开排列，无需其它符号。

关键词：T_EX L_AT_EX xeCJK 模板 排版 论文

EXAMPLE OF BUPT GRADUATE THESIS L^AT_EX 2_E TEMPLATE

ABSTRACT

The Chinese and English abstract should appear after the declaration page. The abstract should present the core of the research work, especially the purpose and importance of the research, the method adopted, the results, and the conclusion.

Key words are terms selected for documentation indexing, which should present the main contributions of the thesis. Key words are aligned at the bottom left side of the abstract content. Key words should be separated by spaces but not any other symbols.

KEY WORDS: T_EX L^AT_EX xeCJK template typesetting thesis

目 录

第一章 绪论	1
1.1 研究背景及选题意义	1
1.1.1 研究背景及国内外研究现状	1
1.1.2 课题主要来源	3
1.2 主要研究内容	3
1.3 论文的结构安排	5
第二章 主题建模介绍	7
2.1 主题模型技术发展概况	7
2.1.1 PLSI 和 LDA 的介绍	7
2.1.2 PLSI 和 LDA 存在的问题	9
2.2 本章小结	10
第三章 分层语义映射的自适应主题生成模型 HLSM	11
3.0.1 复杂网络中社区发现和主题模型的联系	11
3.0.2 分层语义映射 HLSM 模型介绍	11
3.0.3 评估 HLSM 性能的实验	18
3.0.4 文档建模	18
3.0.5 文档分类实验	20
3.0.6 主题模型总结	23
3.1 本章小结	23
第四章 词嵌入 (word embedding) 算法和推荐系统的研究背景	25
4.1 推荐系统简介	25
4.1.1 推荐系统概念	25
4.1.2 近年研究状况	25
4.1.3 现有常见方法	26
4.2 词嵌入 (word embedding) 学习算法介绍	28
4.3 本章小结	32

第五章 主题模型和词嵌入（word embedding）与推荐系统的创新结合	33
5.1 主题模型和词嵌入（word embedding）在推荐系统中应用讨论	33
5.1.1 多场景的冷启动问题	33
5.1.2 长尾数据问题	34
5.1.3 稀疏数据的计算存储规模问题	36
5.2 主题模型在推荐系统中的结合	36
5.3 类 word2vec 词嵌入方法在推荐系统中的结合	40
5.3.1 基于用户行为序列的 word2vec	41
5.3.2 基于商品相似度图生成商品词向量	44
5.4 主题模型和词嵌入在推荐系统中应用实验	44
参考文献	44
附录 A 不定型（0/0）极限的计算	47
致 谢	49
攻读学位期间发表的学术论文目录	51

符号对照表

$(\cdot)^*$	复共轭
$(\cdot)^T$	矩阵转置
$(\cdot)^H$	矩阵共轭转置
\mathbf{X}	矩阵或向量
\mathcal{A}	集合
$\mathcal{A} \times \mathcal{B}$	集合 \mathcal{A} 与集合 \mathcal{B} 的 Cartesian 积, 即 $\mathcal{A} \times \mathcal{B} = \{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}\}$

第一章 绪论

1.1 研究背景及选题意义

1.1.1 研究背景及国内外研究现状

随着时代的发展，我们积累了前所未有的大规模的文本数据。在这些文本数据中包含了大量有价值的信息。如何在海量的数据中寻找出文字间的语义关系已经成为了许多领域的重要问题。如搜索领域如果直接计算文档相似度那么会面临无法承受的计算量同时会受到语义漂移的影响。主题模型可以为每一篇文档分配一个主题，为处理海量的文档集合提供了一个更有前景和更能拓展的方法。

主题模型是一种对文本信息隐含主题进行建模的方法，目标是从文本信息中提取出隐含的主题来标识文本的语义信息。对比传统的信息检索中用到的文档相似度计算方法，主题模型所提取出的信息不仅仅是文档间词分布的相似，还提取出了更深一层更具泛化性的语义信息。基于这些主题，我们可以解决跨域文本分类^[?]，理解文本聚类^[?]，文本推荐^[?]和其他相关文本数据应用的问题。

这些年来各个方向的研究人员们已经对从语料库进行的无监督学习进行了全面研究，潜在主题模型在现有方法中发挥了核心作用。主题模型根据能观测到的单词在文档中的分布从语料库中提取潜在的主题分布，由这些主题分布代表潜在语义空间中的文档。这种潜在语义空间的表达弥合了文档和单词之间的差距，从而实现了语料库的高效处理，如浏览，聚类和可视化。**PLSA**^[?] 和 **LDA**^[?] 是两个众所周知的文档建模主题模型，将每个文档看做是不同主题 (topic) 的混合组成，同时每个文档的生成过程也是在这些主题中根据单个主题 (topic) 内的单词 (word) 分布来选择单词。

目前经典的主题模型 PLSA，LDA 已经被各个领域所广泛应用。不论是学术界还是工业界，主题模型都取得了不错的成果。但是大部分的研究工作都是集中在对 PLSA 和 LDA 这种生成式模型的扩展和优化上。这类生成模型通过对潜在分布的建模来解释观测到数据的分布，比如文档方面可以解释为什么有些部分是相似的。最关键是 PLSA 和 LDA 对于主题分布有一个比较漂亮的数学解释，其实 PLSA 和 LDA 本质上就是做了一个在概率空间上，对于文档 (doc) 和单词 (word) 共现矩阵的一个矩阵分解，这个矩阵分解的基就是主题分布。那么这些主题分布可以用于文

档的分类，文档语义相似度的计算，以及单词（word）的主题分布本身也能作为一种词向量，词表示。可以用于自然语言处理各个方面研究上。

在主题生成这个问题上，可观测到的数据是每一个单词和文档的共现关系（即一个单词（word）在文档（doc）中出现一次算作（word）和（doc）共现一次）。生成模型把整体的单词（word）和文档（doc）的共现矩阵看做是两个相对独立的多项分布 $p(w|t)$ 和 $p(t|d)$ 的乘积，其中 w 代表单词， d 代表文档， t 代表主题。对于 PLSA 和 LDA，他们的优化目标都是找到一个 $p(w|t)$ 和 $p(t|d)$ 的分布使得整体的似然度最大。但是这个优化问题整体是一个非凸的优化问题，因此会存在非常多的局部最优解。Blei 在 2010 年的研究表明，现有的优化方法都存在一个问题，即对于同一份数据集，用同样的一种优化算法，以最大似然为目标分别运行多次，最终会获得不同的模型。

同时对于 PLSA 和 LDA 都有一个很大的问题，主题数 K 是需要预先人工设定好的，并且最终的效果和主题数 K 密切相关。但是主题数 K 是很难通过先验知识得到的，对于主题数 K 有许多相关研究，但是并未提出有效的方法。目前一个主流的做法是，扩大主题数 K ，因为主题数 K 相当于拟合模型空间的秩，秩越大模型的表达能力就越强。但是 Blei 2010 年的研究也表明一个更大的主题数 K 之下推测得到的模型效果并不能比一个合适的主题数 K 之下推测得到的模型效果好。

因此本文希望能探索出一种新的主题建模学习方式，解决主题数 K 在以往被当做超参数的问题，同时这个超参数对最终结果影响非常明显。

并且主题模型的应用很多时候被局限在了自然语言处理领域的一些子任务上，其实当把词和文档的意义平移到其他领域也能有效的运用主题模型，本文会尝试把主题模型平移到推荐领域。

同时在对语料进行无监督学习的整个方向中，衍生出了另一个方向，词嵌入式表达（word embedding）学习。词嵌入式表达^[? ? ?]（word embedding）学习的思路是通过一些和最终任务不是直接关联的监督信息，如单词（word）的 N-gram^[?] 信息，来学习单词（word）或者文档（doc）在潜层语义空间中的表达。而这个思路从观测量—单词在文档集各个文档中的分布，最终推断的隐藏量—单词和文档的潜层语义空间表达，都是相通的。在自然语言处理领域有工作将两者结合起来，比如将主题模型的结果作为词嵌入式表达（word embedding）学习的先验知识来引导其训练。本文不会去具体探究在自然语言处理子任务上的优化，而是试图探索两者在商品推荐领域的一些应用和创新。

自硬件上面的发展取得质变后，通信和计算效率的提高为后续互联网的蓬勃发

展提供广阔的空间。当下互联网的各种应用充斥在人们各种各样的生活中，我们的生活在每一天都面临着各种各样的选择，看什么电影，去哪家餐馆吃饭，住哪一家旅店。这些决策的候选集非常之大，给人们带来很大的困扰，人们要做出这些决策常常是非常痛苦的。如果有一个系统能帮助人们极大的缩小整个候选范围，这个系统无论是促进消费者更快捷的消费，还是在帮助用户找到自己更需要的商品上，都是非常有意义的。因此在当下推荐系统也是非常热门的一个研究领域，也有着非常多的研究成果。

诚然，主题模型和词嵌入式 (word embedding) 学习应当被归属于自然语言处理 (NLP) 领域，很多时候被大家应用在信息检索^[? 1]，或是信息判别等各个领域。但是有一个被大家忽视的点在于，无论是信息检索领域还是推荐系统，要解决的问题都很类似，即信息筛选问题，从广袤的信息中抽取出真正被需要的信息，而主题模型和词嵌入 (word embedding) 技术的问题源头也是这样，都是为了更好的表达出每篇文档的语义。既然问题的出发点是同源的，其应用上就不应当彼此隔离。这些年来主题模型和词嵌入 (word embedding) 的技术在各个自然语言处理 (NLP) 子任务上都取得了长足的进展，然而在推荐系统领域的应用还几乎没有。因此将主题模型和词嵌入 (word embedding) 等自然语言处理领域的核心技术良好的应用到推荐系统领域，也是一个不错的创新思路。本文会仔细分析推荐系统于主题模型以及词嵌入 (word embedding) 的关系，并将这些技术应用到商品推荐系统上。

1.1.2 课题主要来源

本论文的课题研究主要基于对主题模型的研究创新提出了一种新的主题模型 HLSM，同时与世界最大的电商平台-淘宝网合作，研究了主题模型和词嵌入技术在商品推荐领域的应用，所有的实验数据都是工业级的真实数据，验证效果的数据也是真实场景下的数据。

1.2 主要研究内容

本文研究的主要内容集中在主题模型领域，试图探索出一种新的根据语料库自适应的主题建模方法，能够提高整个模型对文档的表达效果，为主题模型提供一种新的思路。同时聚焦于主题模型和词嵌入式表达 (word embedding) 学习这些对文档集的无监督学习方法，它们都试图于获得语料集合的潜层语义空间表达，从而应用到其他的自然语言处理子任务上。本文会分析其和商品推荐问题的关联，探索这些

方法在商品推荐领域的应用，并用实际的工业级的数据来客观实验，验证想法和最终效果。

具体的研究工作如下：

一、提出新的自适应主题建模方法

PLSA 和 LDA 在解决主题生成的问题上，都需要人工设定一个主题数 K ，而且模型的效果和这个主题数 K 密切相关。而大多数情况下主题数 K 是无法通过先验知识得到的，只能通过人为的迭代尝试得到一个比较好的解。在许多对主题模型的研究中都提到了寻找主题数 K 的问题，然而目前并没有一个很好的解决方案。

事实上，PLSA 和 LDA 都认为文档之间的不同主题中单词的分布 $P(word|topic)$ 是相互独立的，因此主题数 K 其实可以看做是主题建模求得主题概率分布的空间的秩，即这个解空间的自由度，不巧的是各个研究者的实验都表明这个自由度对最终模型效果的影响非常之大。因此人工的选择是非常不合理的方案。

如果我们将问题抽象到另一个角度，把所有词连接成一个网络，主题生成的过程其实类似于复杂网络中社区发现的问题。在物理学的研究领域，复杂网络社区分析方面有很多有效的研究可以自动发现社区个数与社区。因此希望借鉴社区发现的思想来自动发现主题个数与主题。同时之前的主题模型在主题粒度上并没有做特殊的优化，有一些关于分层主题的研究但是有效的成果比较少，主题模型应当对高层抽象的主题和底层具体的主题有不同的处理。本文综合这些方面的考虑，提出了一个新的自适应分层主题建模的方法，可根据语料库中能观测到的词语分布信息，自适应的发现主题个数，并且在具体层级的主题和抽象层级的主题上做出区分，以及抽象主题包含具体主题的关系。

二、探索主题模型和词嵌入式表达 (word embedding) 等潜层语义方法在商品推荐领域的应用

深入的分析整个主题模型作用的过程和原理也是一个重要的过程，因为这个本质了解清楚之后可以将这些理论泛化到其他的问题，而不是仅仅局限于语言信息处理领域。同时词嵌入式表达 (word embedding) 和主题模型的本质观测空间是相似的，都是这些被处理的语料库中所有单词在文档中的分布情况，细小区别是有的词嵌入式表达 (word embedding) 方法如 word2vec^[2] 的观测粒度更细，有窗口的概念。同时其希望推断的隐藏变量也是相似的，主题模型是概率空间约束下的文档和单词的表达，而词嵌入式表达学习 (word embedding) 是一个对单词的向量空间表达，同时未明确约束其物理意义。

而商品推荐领域中研究的一个主要问题也是用户 (user) 和商品 (item) 的一个关系。如果我们能寻找到能包含这种关系信息的其各自向量空间表达。这对于商品推荐系统的最终推荐效果和其本身在不同场景解决问题迭代的效率都会有巨大的贡献。

目前的生成式的主题模型，其实是做了一个概率空间上，对文档 (doc) 和单词 (word) 共现矩阵的分解，分解矩阵的基是主题分布。而这个意义和推荐常用的协同过滤矩阵分解是相似的。而 word2vec 也是类似，对文档 (doc) 和单词 (word) 的 PMI 矩阵的分解。其实把用户 (User) 对商品 (Item) 的行为链映射为文档 (doc)，商品 (Item) 映射为词 (word)，就可以简单的把商品推荐和主题模型结合起来。

本文将传统主题建模的思路和词嵌入式表达 (word embedding) 的思路抽象到了商品推荐领域，值得注意的是本文的目的不仅仅是将主题模型和词嵌入 (word embedding) 技术做一个应用领域的迁移，而是会仔细的分析推荐系统和这两个技术的关系，以及为什么这些技术会在推荐系统有效。后文中会仔细介绍如何产生这个想法以及这个思路要解决的问题，并用工业级的数据做了客观的实验，验证了这方面想法，以及其具体实验效果，并且最终证明了其有效性，最后在真实的商品推荐系统上上线了这一模型。

研究生期间，本人在“主题建模”课题上的主要研究成果包括：

- 以学生一作身份发表 EI 论文一篇。

1.3 论文的结构安排

全文内容安排如下：

第二章 主题建模介绍

本章将总结一下已有的主题模型方法，以及其发展概况，并探讨其目前仍未解决的问题。

2.1 主题模型技术发展概况

主题模型从一组文档生成主题，并将主题分配给这些文档。基于这些主题，我们可以解决跨域文本分类^[1]，理解文本聚类^[2]，文本推荐和其他相关文本数据应用的问题。虽然对主题模型存在着非凡的研究，但大多数研究集中于基于概率潜层语义分析 (PLSA)^[3] 和潜层狄利克雷分布分析 (LDA)^[4] 改进的生成模型。

2.1.1 PLSI 和 LDA 的介绍

主题建模中的最先进的方法是去尝试拟合文集中的文档的生成模型的参数的值。研究生成模型的第一个主要尝试是概率潜层语义分析 (PLSA)^[3]。目前的黄金标准是潜层狄利克雷分布分析 (LDA)^[4]。第一次，主题模型提供了一种用于聚类文本文档的原则性方法，具有一组明确的假设。这种方法刺激了一系列的研究，旨在推广模型和放松他们的假设。

基于 PLSA 和 LDA 算法的生成模型假设每个主题由特定词使用概率分布表征，并且语料库中的每个文档是从主题组合体生成的。作为示例，考虑从两个主题，数学和机器学习领域生成的文档的语料库。图 2-1 语料库中的每个文档将从具有特殊概率的主题集中依据这些主题分布抽取单词组成文档。例如，一个文档 d_{math} 主要来自于数学主题的概率是 $p(topic = math|d_{math}) = 0.8$ 来自于机器学习的概率是 $p(topic = ml|d_{math}) = 0.2$ 。

具有不同主题组合体的文档将使用不同的词，因为使用给定词的概率分布取决于主题。重要的是，假设一些单词将与单个主题密切相关，否则，将不可能拟合该模型。例如，诸如“loss”或“optimization”的词将主要用于机器学习聚焦的文件中，因为 $p(word = loss|topic = machinelearning) >> p(word = loss|topic = math)$ 。相比之下，诸如“evolution”或“equation”的词将主要用于数学聚焦的文档中，因为 $p(word = evolution|topic = machinelearning) << p(word = evolution|topic = math)$ 。然

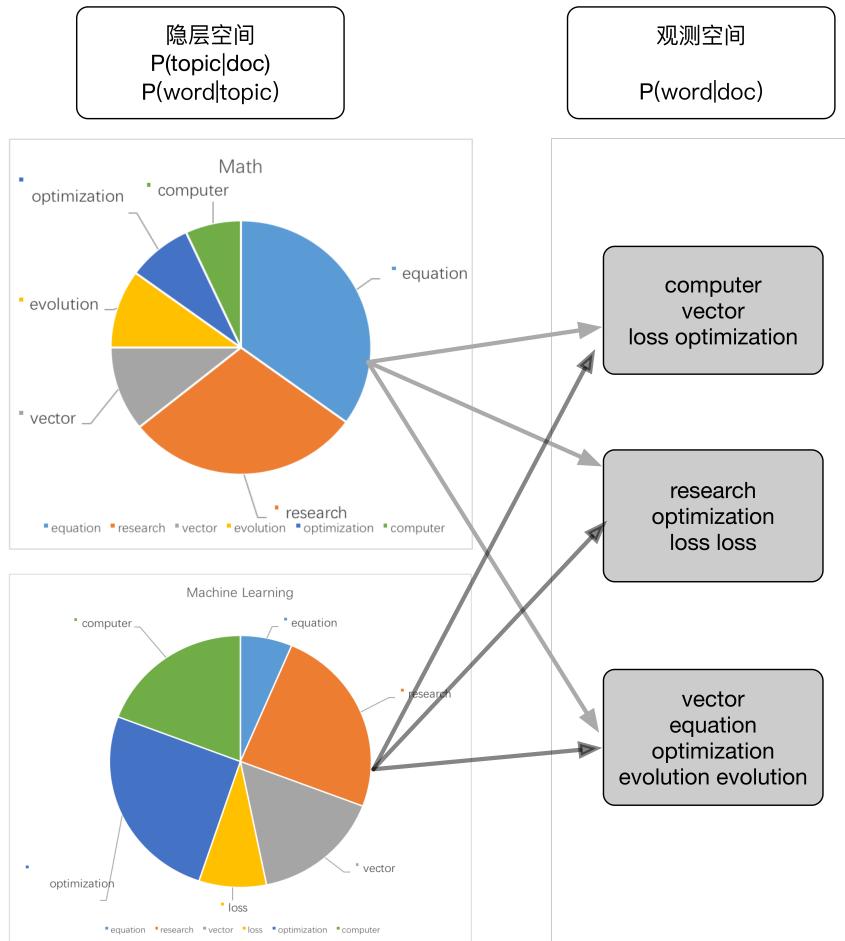


图 2-1 文档中的文档的生成模型假定是主题的组合体。主题结构是潜在的，意味着人们不能访问用于在语料库中生成文档的“真实”主题集合。然而，可以使用主题模型算法来估计主题集合的分布。为此，在该示例中，计算每个文档中的词频率并将它们建模为不同主题，数学和机器学习的组合体。

而，会有其他词，如“research”或“vector”，是通用的，并且两个主题几乎相同。在实践中，人们只能访问每个文档中的字数，而实际的主题结构是不可观察的，即潜在的。因此，挑战是估计主题结构，其由概率集合 $p(\text{topic}|\text{doc})$ 和 $p(\text{word}|\text{topic})$ 来定义。

为了具体的解释这个问题，我们假设由 N 个文档组成的语料库使用 N_w 个不同的单词从 K 个主题生成。然后，需要估计 $N \cdot K$ 个概率 $p(\text{topic}|\text{doc})$ 和 $K \cdot N_w$ 个概率 $p(\text{word}|\text{topic})$ 。PLSA 和 LDA 都旨在估计具有产生数据的最大化似然可能性的这些 $K \cdot (N + N_w)$ 个概率的值^{[3][4]}。因此，PLSA 和 LDA 依赖于非线性地依赖于大量变量

的非线性最大化，即非确定性多项式时间难问题。

两个模型之间的主要区别在于，对于 PLSA， $N \cdot K$ 个概率是自由参数，其必须直接从数据中估计，而 LDA 假设概率集合是从狄利克雷分布中抽取的随机变量 [34]。因此，对于 LDA，只需要估计 K 个参数（每个主题一个） $\alpha_1; \alpha_2; \dots; \alpha_K$ 。这些 α 称为超参数。LDA 的参数数量通常被进一步减少，假设所有超级主体采用相同的值，通常由 α 表示。假定超级单元的单个值的 LDA 的“版本”被称为对称 LDA，而具有 K 个超级单元的完整模型被称为非对称 LDA^[5]。

2.1.2 PLSI 和 LDA 存在的问题

2.1.2.1 先验的敏感性

如上一节中所解释到的，LDA 和 PLSI 都旨在依据产生语料库的最高可能性来估计概率分布 $p(topic|doc)$ 和 $p(word|topic)$ 。因此，推理问题被转换为优化问题^[6]。但是对于同样的文档集存在许多具有几乎相同的可能性的竞争模型。由于似然度的高度退化，标准优化算法将更有可能在不同的优化运行之后推断不同的模型，而不是推断具有最高似然性的模型。因此对于 LDA 先验尤其重要。在 LDA 研究的前沿文章^[7] 中，深入的讨论了先验对 LDA 的影响，并做了相当客观的实验证明了其结论。其实验表明非对称的 LDA 性能优于狄利克雷先验参数用均匀初始化的 LDA，然而，值得研究为什么这种非对称的先验提供了卓越的性能。

主题建模的主要假设是主题应该捕获语义相关的词共现。主题也必须是独特的，以传达信息：知道只有少数同时出现的单词应足以解决语义环境。因此，我们不希望特定主题在单词上的分布与任何其他主题类似。因此，非对称先验 α 超过均匀的 α 是一个坏主意：基本量度将反映语料库的词使用统计，并且先验地，所有主题将展现这些统计。一个对称的先验 α 仅仅使得一个先前的语句（由浓度参数 β 确定）关于主题是否将在单词上具有更稀疏或更均匀的分布，因此主题可以自由地作为必要的独特和专门化。然而，仍然需要考虑幂律词的使用。这样做的一种自然方式是期望在给定语料库中的每个文档中，某些单词组将比其他单词出现得更频繁。例如，单词“模型”，“数据”和“算法”可能出现在机器学习会议上发表的每篇论文中。这些假设自然地导致我们经验确定为优越的先验的组合：用于跨文档共享共享性的非对称狄利克雷先验，以及用于避免主题之间的冲突的对称狄利克雷。

虽然 Wallach 等人的研究结果找到了更好的先验方法提高了 LDA 的性能，即非对称 LDA。但是其文中也提到了我们唯一的观测量单词的分布提供的信息并未被

LDA 这个模型充分的使用。

2.1.2.2 主题数目 K 的选择问题

与此同时，选择主题的数量 K 是有限主题建模中最有问题的建模选择之一。对于迄今为止的 K 的各种值，没有用于选择 K 或评估保持数据的概率的有效方法。并且 LDA 对于 K 的不良设置是鲁棒的程度尚未被充分理解^[7]。理想情况下，如果 LDA 有足够的主题来对数据集进行良好建模，则 K 的增加不会对令牌到主题的分配造成影响。即，应该以低频率使用附加主题。例如，如果二十个话题足以准确地建模数据，则通过将话题数目增加到五十个，所推断的话题分配将不会受到显着影响。如果是这种情况，使用大的 K 将不具有对推断的改进。换句话说，我们仍然需要一个健壮的 K 。实际上， K 可以看作是主题生成的解决方案空间的秩。设置 K 等价于人工选择解空间的秩，这显然是不合理的。

虽然 LDA 和 PLSA 之类的图形模型已经催化了无监督学习的大量研究，并取得了许多实际成功，但重要的是要注意，大多数图形模型文献都集中在参数模型上。特别地，图形和包括图形模型的局部势函数被视为固定对象；它们在结构上不增长，因为观察到更多的数据。因此，尽管非参数方法主导了监督学习的文献，但参数方法在无监督学习中占主导地位。这似乎令人惊讶，因为无监督学习问题的开放性质似乎特别与非参数哲学相称。但它反映了无监督学习的基本张力，以获得一个良好的学习问题，有必要强加假设，但假设不应该太强烈，或他们将通知发现的结构，而不是数据本身。

而近期主题模型方向的研究中，研究主题数目 K 依然占有相当大的比重，研究者逐渐开始结合数据本身设计先验来试图找出在推断主题模型之前找出合适的主题数 K 或者是找出有效评价主题数 K 的方法。

2.2 本章小结

本文在此章节主要简略的介绍主题模型方向研究的发展过程，之后深入的讨论了主题模型研究需要解决的问题，以及本文对此的一些观点。介绍部分介绍了主题模型的思路源头和要解决的问题并简略介绍了经典的 PLSA 和 LDA 方法，在后面的讨论部分介绍了主题模型的先验敏感性和主题数目 k 的选择问题，这些问题时本文后续提出新的主题建模思路的拟解决目标。

第三章 分层语义映射的自适应主题生成模型 HLSM

本章在总结了之前的传统主题模型方法的优点和存在的问题后，借鉴复杂网络分析算法映射方程 (MapEquation)^[8] 的思路，提出了一种新的自动主题生成方法，将重点介绍新的主题生成方法的思路并解释其意义。

3.0.1 复杂网络中社区发现和主题模型的联系

所有主题模型算法的核心是找到适合的概率分布 $p(topic|doc)$ 和 $p(word|topic)$ 使得生成文档的似然度最大，即找到以许多局部最大值为特征的似然函数的全局最大值。这种优化问题也是物理系统中无序系统研究的核心问题^[9-11]。

而本质上 $p(word|topic)$ 是主题模型推断的核心， $p(word|topic)$ 是每一个主题下单词的分布，如果我们一开始有一个硬的分布约束，即每个词只能属于一个主题。这样的约束下不同的主题 (topic) 之间不会存在相同的单词 (word)。如果我们将所有的单词 (word) 看做是一个个节点，并将单词 (word) 之间用我们可以从数据中观测到的信息链接起来，那么推断其归属于某个主题的过程，就是一个复杂网络中社区发现的问题。是一个当然这是一个非常强的约束，我们的主题生成并不会就此结束，后续会讲解之后的优化过程。

3.0.2 分层语义映射 HLSM 模型介绍

分层潜在语义映射模型 (HLSM) 是一种用于主题建模的网络方法。类似于众所周知的主题模型，每个文档被表示为对潜在主题的组合体。区分 HLSM 模型和现有主题模型的关键特征是 HLSM 直接将单词进行分块归属到某一个社区中，并将每个社区定义为初始主题，然后细化这些初始主题，因此 HLSM 是以一种全新的方式来推断概率分布 $p(word|topic)$ 与 $p(topic|doc)$ 。HLSM 模型按以下步骤推断主题：

步骤 1. 构建单词连接网络。依据文档集中可观测到的各个文档和单词的共现矩阵构建以单词 (word) 为节点的带权网络。我们通过单词在各个文档中的分布，计算所有至少在一个文档中共同出现过的两个词 (单词对) 之间的关联程度值作为其连接的权值。然后，我们将其中单词之间连接权值在一定阈值以上的连接线保留，其余的减枝。

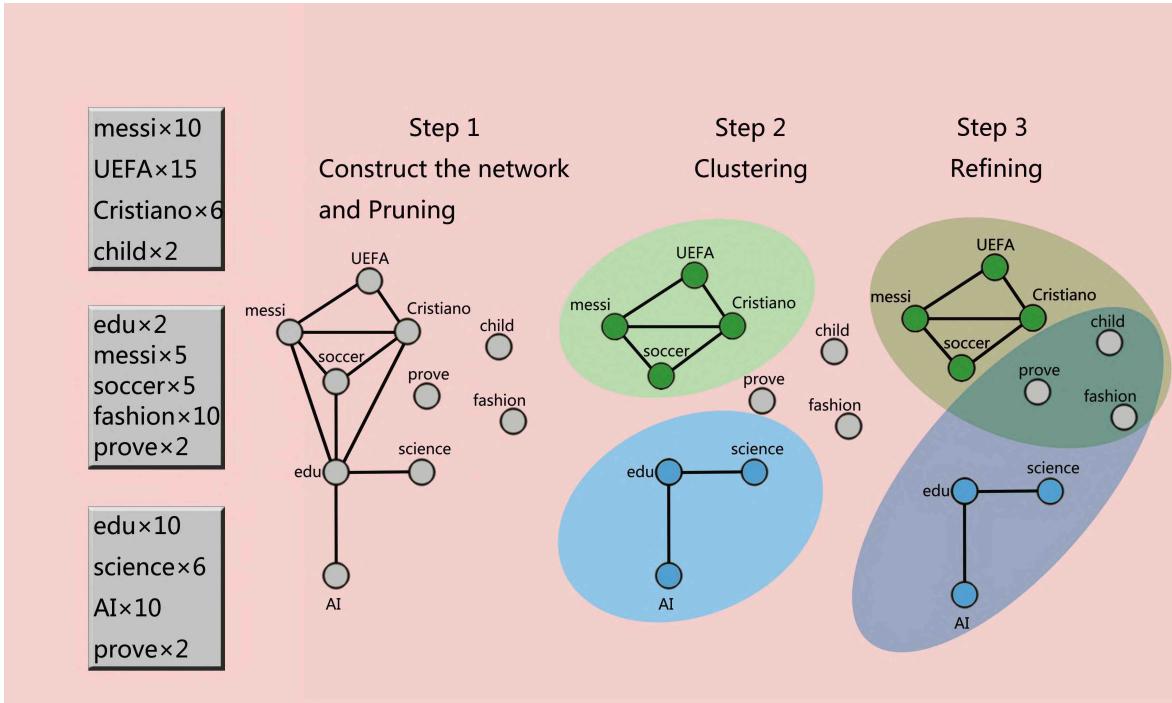


图 3-1 Illustration of the HLSM algorithm.

步骤 2. 分层地对单词进行社区归属。通过在潜在主题空间中的关联来连接整个文档集中的单词。自然地，我们假设语料库中的主题将引起网络中的单词的社区。因此，我们使用分层信息映射方程 (Hierarchical Map Equation)^[12] 来检测社区。在大多数语料库中，主题以多级抽象的形式出现。抽象主题下又包括几个具体主题。因此，我们检测到对应于抽象主题的一些大规模社区，然后我们检测来自大规模社区的小社区，这对应于更为具体的主题。我们采用社区作为对用于生成文档的每个主题的主题数量和单词组成的初始猜测。值得注意的是，我们不人工设置总共的层级数量每个层级社区的数量。分层信息映射方程可以自动揭示单词网络中的多级组织。

步骤 3. 优化初始的概率估计。在最后级别的单词聚类之后，我们可能会发现一些社区和别的社区都没有连接，并且在步骤 2 中，也可能得到可以得到不在网络中的单个单词。并且，合理的主题模型不应该所有的单词都只能属于一个主题，单词因此，在步骤 2 中检测到的先前主题是非常粗糙的，我们使用类似 PLSA 的似然优化来改进这些粗糙的初始主题。

3.0.2.1 构建单词连接网络

词之间的关联权值必须与主题密切相关，以确保基于该网络对词进行社区归属的有效性。但主题是潜在的，所有的观察都是收集到的文档中的单词的分布。现在假设我们用之前的人类知识人工地分配主题，人们可以观察到共享相同主题的文档会有更大的概率共享一些单词。自然地，我们可以认为在许多文档中共同出现的词语共享相同主题，换而言之，这些词语在潜在主题空间中更为相似。

为了计算潜在主题空间中的单词之间的关联，与潜在语义分析（LSI）的核心思想一样，我们基于共生矩阵 M 的奇异值分解（SVD）将单词映射到降维的向量空间，其中每一行 i 对应于一个单词，每一列 j 对应于一篇文档，并且每个矩阵条的元素 M_{ij} 对应于文档 j 中单词 i 的出现次数。整个过程开始于标准的 SVD：

$$M = U\Sigma V^t, \quad (3-1)$$

对角矩阵 Σ 包含 M 的奇异值。 M 的近似值是通过将 Σ 中的所有最大 K 个奇异值设置为零来计算的 ($=\tilde{\Sigma}$)，其意义是在 L_2 范数空间上秩为 K 的表达。

现在获得了 M 的一个近似矩阵

$$\tilde{M} = U\tilde{\Sigma}V^t \approx U\Sigma V^t = M, \quad (3-2)$$

原始高维矩阵 M 是稀疏的，但是相应的低维潜在向量通常不是稀疏的。这意味着我们可以更好的通过这些低维空间计算潜在主题空间中的单词对之间的有意义的关联值，而不是简单的将单词对之间的共现（在同一个文档中出现）次数当做被构建的单词网络的权值。在 HLSM 中，我们计算 $U\tilde{\Sigma}$ 的行之间的余弦距离作为潜在主题空间中每对词的关联值，并且将词 i 和 j 与该关联连接 $S(i, j)$ ：

$$W = U\tilde{\Sigma}, S(i, j) = \frac{\langle W_i \cdot W_j \rangle}{\|W_i\| \cdot \|W_j\|}. \quad (3-3)$$

计算完所有共现单词对的关联值后，我们对整个构建的单词网络进行剪枝。一些共现词对之间的关联值非常低，我们可以假设这些连接是噪声。因此设置 q 的阈值以使关联权值低于 q 的边被除去。

3.0.2.2 分层地对单词进行社区归属

在大多数语料库中，主题的结构并不是简单的单层结构，总是可以有多个层次。一些具体的主题在同一个抽象主题下。例如，集中在“足球”上的语料库中的词语可能来自“名星”，“竞赛”，“足球史”等主题。

我们基于潜在主题空间中的词之间的关联来构造词的网络。如果主题的原始结构是多级的，则网络也应该具有多级结构。要在上一节中建立的单词网络中发现多个级别的社区，我们选择 *Hierarchical Map Equation*^[12]。值得注意的是，我们没有人为了每个级别设置级别数量和社区数量。相反，层次映射方程^[12]可以通过这个模拟信息流在网络中的随机游走，并通过类似贪心的算法，自动揭示单词网络中的多级结构。映射方程（Map Equation）提出了寻找网络中的社区结构和最小化表达随机游走者在网络上的移动的编码长度之间的对偶性。对于给定的网络分区，映射方程定义了在理论上可以描述该随机游走的轨迹的最短编码长度 $L(M)$ 。映射方程的核心思想是，如果随机游走者倾向于长时间停留在网络的一些分区中，则可以通过在这些分区内部的节点上公用一个表达分区的编码，从而减少表达整个网络中每一步游走所需的平均编码长度。其本质，是通过规定了分区的划分，从而减少了整个网络的不确定性，也就是表达整个网络的熵。因此，当用于网络中的实际流随机游走时，在所有分区可能的网络分区上估计最小表达编码长度 $L(m)$ 可以依据网络中随机游走的动力结果揭示网络本身的结构。在主题建模的问题中，对应于 n 个节点的分层网络 M ，每个节点对应一个单词（word），被划分为 m 个分区。在每一个分区中有一个拥有 m^i 个子分区的子映射 M^i 。相应地，在每个子分区 ij 中有一拥有 m^{ij} 个子分区的子映射 M^{ij} ，以此类推。

对应的层级映射方程 $L(M)$ 是

$$L(M) = q_{switch}H(Q) + \sum_{i=1}^m L(M^i) \quad (3-4)$$

中间级别的子映射 M^i 的映射方程

$$L(M^i) = q_{switch}^i H(Q^i) + \sum_{j=1}^{m^i} L(M^{ij}) \quad (3-5)$$

最底层即最细粒度的映射方程

$$L(M^{ij...k}) = p^{ij...k}_{in} H(P^{ij...k}) \quad (3-6)$$

每个码本的权重取决于码本的使用比率，并且 LM 是每个码本的码字的平均长

度的和。 HQ 是根据其使用率的索引码本中的码字的平均长度，而熵项取决于码本被使用的比率。在任何给定步骤上，随机游走者以 q_{switch} 的概率切换第一级分区，而 q_{switch} 是使用索引码本的比率。在每个子分区级别， HQ^i 是根据子索引码本中的使用比率的码字的平均长度，并且 q_{switch}^i 是用于进入 m_i 子分区或退出到更高级别的码字使用比率。在最后一级， $H(P^{ij...k})$ 是对应于子分区码本中的使用比率的平均编码长度， $p_{ij...k_{in}}$ 是子分区 $ij...k$ 中随机游走的代理访问节点或者跳出到其他子分区的比率。找最好地表示结构的分层结构的问题被转换为找到具有最小映射方程的网络的分层分区。图3-2说明了一个映射方程的例子。

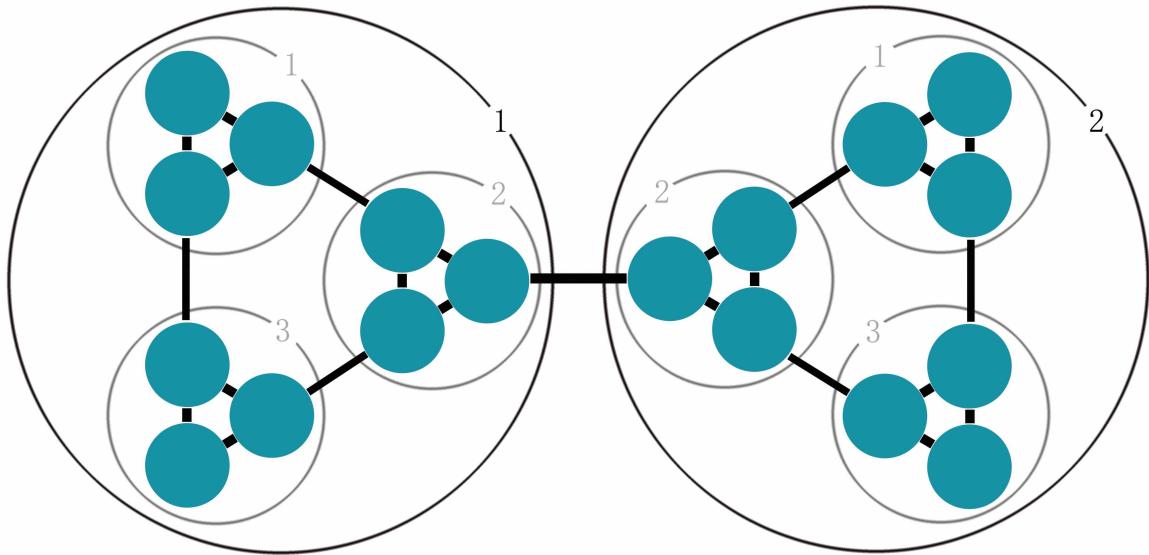


图 3-2 通过最小化映射方程来找到对应于网络中最优分区结构的例子。

在这个例子中，我们可以假定网络中连接的所有权重是相等的，因此所有比率都可以通过计算连接边的数量和归一化来得到。未分区网络的映射方程（平均表达编码长度）为 $-\log_2 1/18 = 4.17bit$ 。在网络被分区后，第一级分区的码字以总比率 $q_{switch} = \frac{2}{50}$ 被使用（考虑移动方向时，网络中有 25 条线路和 50 个可能的移动，而只有 2 个移动可以在第一级分区之间切换），同时第一级的两个分区的使用比率就是 $Q = \frac{1}{2} \frac{1}{2}$ 。同时分区 1 内的子分区使用比率分布 $Q^1 = \frac{2}{8} \frac{2}{8} \frac{3}{8} \frac{1}{8}$ ，注意一点，随机游走的代理有可能以 $q_{switch}^1 = \frac{8}{50}$ 的概率从分区 1 跳转入分区 2。因此 LM 是：

$$L(M) = q_{switch}H(Q) + \begin{cases} q_{switch}^1 H(Q^1) + \begin{cases} p_{in}^{11} H(P^{11}) \\ p_{in}^{12} H(P^{12}) \\ p_{in}^{13} H(P^{13}) \end{cases} \\ q_{switch}^2 H(Q^2) + \begin{cases} p_{in}^{21} H(P^{21}) \\ p_{in}^{22} H(P^{22}) \\ p_{in}^{23} H(P^{23}) \end{cases} \end{cases} \quad (3-7)$$

3.0.2.3 优化初始的概率估计

一旦网络建立，我们使用分层映射方程 (*Hierarchical Map Equation*) 来检测高度关联词的簇（与由分层映射方程 (*Hierarchical Map Equation*) 检测到的分区相同）。在最后一层分区探索完全之后，我们得到一个对单词非常硬的分区，意味着单词只能属于一个单一的分区。而这里的每一个分区对应我们的主题建模问题，是一个初始的主题。实际上，一个词在不同的上下文中可能具有多种意义和多种类型的使用。因此，如果我们简单地将每个分区定义为一个主题，这些粗略的主题不能提供关于语料库对应于潜在主题空间的合理的概率解释。因此，我们提出了一种进一步改进这些粗糙初始主题的方法。

我们现在讨论如何在给定了每个词属于的一个主题（分区）后去计算 $p_{topic|doc}$ 和 $p_{word|topic}$ 。在之前的单词分割中，我们将每个分区定义为一个主题。事实上，在分层映射方程 (*Hierarchical Map Equation*) 处理之后，网络中的每个单词 (word) 都只能位于一个分区中。因此， $pt|w = \delta_{tw}$ 。 $\delta_{tw} = 1$ ，如果单词 w 位于对应于主题 t 的分区中。对于其他主题 \bar{t} ， $\delta_{\bar{t}w} = 0$ 。注意到在这一步中，词 w 只能属于一个主题 t ，所以 $p_{wt} = pw$ ，同时 $p(t|d) = \frac{\sum_w p(t|w)}{L_d}$ 因此：

$$p(w|t) = \frac{p(w)}{\sum_w p(w) \times \delta_{t,w}} \text{ and } p(t|d) = \frac{1}{L_d} \sum_w w_w^d \delta_{t,w}. \quad (3-8)$$

其中 L_d 是文档 d 中的单词数， w_w^d 是单词 w 出现在文档 d 中的次数。 $n(w,t) = L_C \times p(w,t)$ 是在生成整个语料库时主题 t 被选中后，从主题 t 的所有单词中选中单词 w 的次数。 L_C 是语料库中的单词的数量。所以目前为止，我们的主题模型的类似

PLSA 的似然函数为:

$$\begin{aligned} L &= \log\left(\prod_{w,d} p(w,d)\right) = \log\left(\prod_{w,d} \sum_t p(w|t)p(t|d)\right) \\ &= \sum_d \sum_w w_w^d \times \log\left(\sum_t p(w|t)p(t|d)\right). \end{aligned} \quad (3-9)$$

现在文档中的每个词都属于且仅仅属于一个主题，那么整个文档所对应的总的主题数是非常多的，且对应于其中少数词属于的主题，这篇文档可能在这里并不是想表达这个主题的语义。因此我们可以通过简单地使文档中的词更具体的去归属于较少的主题来提高整体生成当前文档的似然度。为此，我们的优化算法简单地为每个文档找到分配有一些不常出现的主题的单词，并将该文档中最重要的主题重新分配给这些单词。

步骤 1. 对于每一篇文档 d , 我们依据最小化 $p-value$ (后面会介绍) 来找到它最重要的主题 t_s 。在主题模型中，每个单词 w 都是独立的从每个主题 t 中采样出来的，反过来主题被选中的概率 $p(t) = \sum_w p(w)p(t|w)$ 。同时我们把 x 定义为文档 d 中从主题 t 中采样出的单词的个数 (主题为 t 的单词的个数)，那么依据前文公式 (modify) $x = L_d \times p(t|d)$ ，此时我们可以通过伯努利分布来计算主题 t 的 $p-value$, $p-value = B(x; L_d, p(t))$ 。显然 $p-value$ 对主题 t 在文档 d 中的重要性的表达能力要强于 x ，因为 x 仅仅和 $p(t|d)$ 相关，而 $p-value$ 考虑了整个语料库的信息 $p(t)$ 。

步骤 2. 在构建单词网络的那一步中，对于每一篇文档 d ，都有可能含有一些单词独立在这个网络之外，也就是和所有词都没有共现或者是权值太低连接线被剪枝掉。对于这些单词，我们很简单的将这篇文档中最重要的主题当做这些单词的主题，由此我们可以计算一个底线的类似 PLSA 的似然度 $L(modify)$ 。

步骤 3. 对于每一篇文档 d ，我们通过设定了一个参数阈值 η 来将那些不常出现的主题 $p(t_{in}|d) < \eta$ 定义为 t_{in} 。同时将前文提到的最重要的主题赋予给这些从 t_{in} 选出来的单词。需要注意的是，所有 $p(t_{in}|d)$ 都会被置为 0，与此同时 $p(t_s|d)$ 会获得所有 $p(t_{in}|d)$ 的和的增长。相应的， $n(w, t_{in})$ 会减少相对于那些在文档中属于 t_{in} 的单词的数量 w_w^d ，而 $n(w, t_s)$ 获得相应的提高。

步骤 4. 在前面的步骤依次在所有的文档都执行完之后，我们开始计算:

$$p(w|t) = \frac{n(w,t)}{\sum_w n(w,t)} \quad (3-10)$$

同时整个模型的似然度， L_η ，在这里和我们定义的参数 η 相关。如果最终的目的是要获得最好的模型并且在可以进行交叉验证的场景下，我们简单的通过将 η 从 0% 到 50% 以每步 1% 的方式迭代验证，从中一局最大的似然度 L_η 来选择最终模型，这里是一个很简单的参数寻优过程，只用了简单的交叉验证来寻找并未探索更看起来简便的方法，不过由于后面的迭代计算量并不大，所以对模型的计算开销不大。

HLSM 从训练数据中去预估两个概率分布 $p(w|t)$ 以及 $p(t) = \sum_w p(w)p(t|w)$ ，对于一个新的文档，HLSM 的推断过程非常简单，固定 $p(w|t)$ ， $p(t|d)$ 可以通过如下公式计算：

$$p(t|d) = \frac{\sum_w p(t|w)}{L_d} \quad (3-11)$$

由于在训练过程（更严格的说法是在训练数据集上推断完概率分布）之后，HLSM 固定了 $p(w|t)$ 和 $p(t)$ ，因此 HLSM 存在较大的过拟合风险，这个在训练集数据比较小的时候会是 HLSM 的一个缺点。

3.0.3 评估 HLSM 性能的实验

HLSM 是一个面向语料库处理的主题模型。它可以应用于许多方向的应用，如文本分类，聚类，过滤，信息检索等相关领域。我们在这里按照 Blei 的想法^[4]，在本节中，我们考察 HLSM 和其他传统主题模型在两个重要的应用场景上的应用：文档建模和文档分类。

3.0.4 文档建模

文档建模的根本目的是将训练的模型从训练数据集推广到新的未见过的数据集。语料库中的文档是未标记的，我们的目标是对文档的主题概率分布进行估计，因此我们希望在训练集上表现良好的模型在测试集依然能获得较高的似然度，这样才能证明模型的可推广性。具体来说，我们计算了一个测试集上的混淆度来评估模型的性能。产生较低混淆度的模型被认为实现更好的泛化性能，因为这样的建模方法获得的模型在处理该模型从未见过的那部分数据集时不会显得困惑。正式地，对于 M 个文档的测试集，混淆度被定义为：

$$\text{perplexity}(D_{test}) = \exp \left\{ \frac{-\sum_{i=1}^M \log p(d_i)}{\sum_{i=1}^M L_i} \right\} \quad (3-12)$$

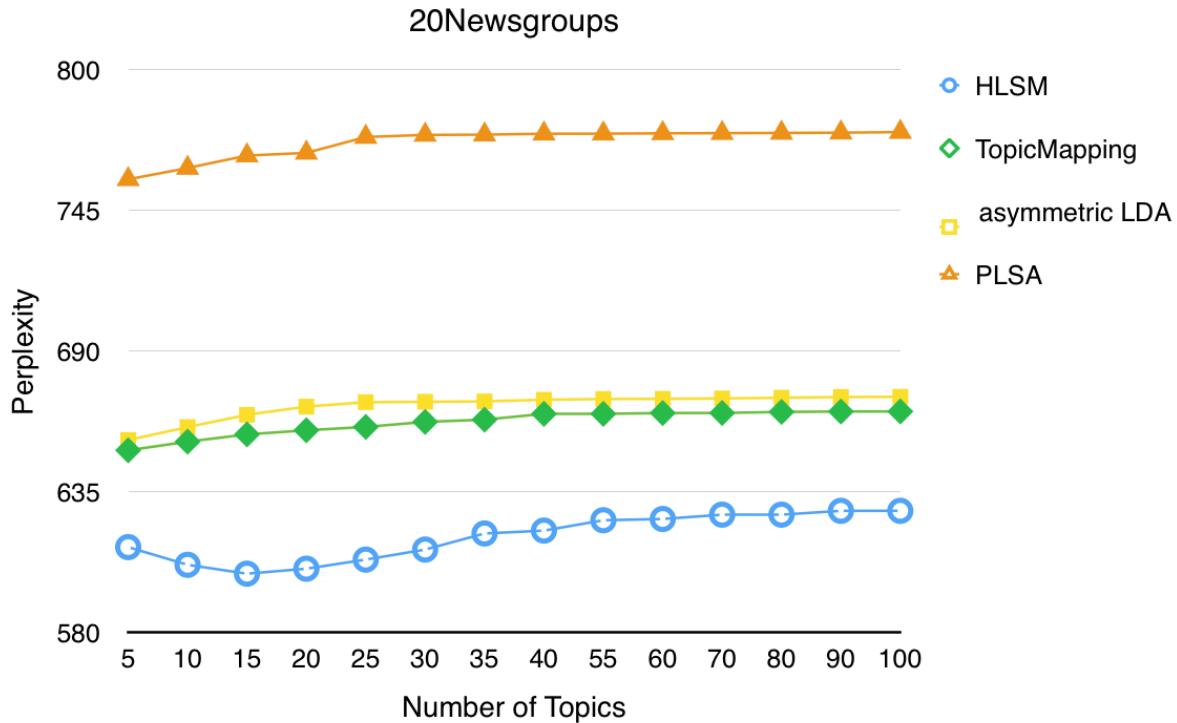


图 3-3 Perplexity comparisons on the 20Newsgroups dataset.

我们在 20Newsgroups 数据集的一个子集上进行了这个实验，该数据集已被广泛用于评估跨域文本分类算法的性能。它包含近 20,000 个新闻组文档，这些文档已被均匀分为 20 个不同的新闻组。我们选择了 3878 个文档（我们过滤了一些小文档），来自四个领域：comp.graphics, com.sys.mac.hardware, sci.crypt 和 sci.med 作为我们在评估中使用的数据集。我们随机抽取了 20% 的语料库用于测试，并在剩余的 80% 文档中做训练。在数据预处理方面，我们从标准的原始文档中删除了 163 个停用词，并从每个语料库中删除了出现次数小于 3 的单词。我们比较了 HLSM 与 PLSA, asymmetric LDA (非对称 LDA) 和 TopicMapping 等模型的效果。对于所有主题，asymmetric LDA 的初始 α 设置为 0.01。

图3-3显示了主题数量从 5 变化到 100 不同模型的混淆度结果。这里由于 HLSM 是一个自动寻找到主题数的模型，我们没有去过多调节上一节中提到的唯一的超参数 η ，而是在计算混淆度的时候如上一节中的迭代优化过程一样循环的将那些在文档中占比小的主题所属词归属到最重要的几个主题内，从而计算 5 到 100 个主题数的混淆度。可以看出，HLSM 模型在混淆度方面实现了轻微的改善，而 TopicMapping 接

近于不对称 LDA。实验表明，HLSM 依靠对单词网络构建后的层次分区发现得到的先验猜测在主题生成的性能上有较明显的提高。

表3-1提供了 HLSM 在数据集 *Comp & Sci* 上提取的 $p(t|d)$ 最大的 12 个主题的示例，一些具有较低概率的主题未被展示出来。我们用学习的主题词概率 $p(w|t)$ 对词进行排序。通过检查主题词，我们可以观察到同一主题中的词总是语义相关的。例如，主题 1 是关于 Mac 硬件，而数据集 *Comp & Sci* 中的一个领域正好是 comp.sys.mac.hardware。值得注意的是，一些主题在抽象层面看起来相似，但它们之间在更具体的意义上仍然有一些区别。例如，主题 2 和主题 4 中的单词在语义上相关，但主题 2 与医疗更相关，而主题 4 可能描述了一些关于疾病的报告。结果表明，我们的方法可以有效地识别来自不同领域的领域特定功能之间的相关性。此外，我们的方法可以提取抽象领域范围内的更为具体或者说更细粒度的主题。我们还在 20Newsgroups 数据集进行了下一个实验。

表 3-1 Comp and Sci 数据集上 HLSM 生成的 Top12 个主题的单词分布

topic: 1 $p(t) : 0.080127$	topic: 2 $p(t) : 0.06720$	topic: 3 $p(t) : 0.066154$	topic: 4 $p(t) : 0.061912$	topic: 5 $p(t) : 0.060678$	topic: 6 $p(t) : 0.060007$
mac	doctor	clipper	medic	food	imag
doe	patient	phone	health	msg	jpeg
system	vitamin	chip	1993	diet	file
speed	medic	encrypt	diseas	eat	format
price	candida	govern	hiv	weight	gif
hardware	treatment	onli	report	effect	program
topic: 7 $p(t) : 0.056193$	topic: 8 $p(t) : 0.055702$	topic: 9 $p(t) : 0.050720$	topic: 10 $p(t) : 0.046258$	topic: 11 $p(t) : 0.045494$	topic: 12 $p(t) : 0.044068$
imag	drive	key	anonym	nsa	3d
data	disk	encrypt	email	writes	graphic
system	system	messag	internet	govern	file
packag	work	secur	post	articl	object
sourc	scsi	pgp	comput	david	ray
code	machin	attack	inform	trust	model

3.0.5 文档分类实验

在文本分类问题中，主题模型希望将文档划分成两个或更多个相互排斥的类别。特征的选择是文档分类问题的一个具有挑战性的问题。通过以潜在主题空间表示文档，主题模型可以生成概率 $p(t|d)$ 。如果使用 $p(t|d)$ 的向量作为文档的特征来解决文

表 3-2 从 20Newsgroups 抽取出的数据集

Data set	Domain
Comp and Sci	comp.graphics comp.sys.mac.hardware sci.crypt sci.med
Comp and Talk	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware talk.politics.mideast talk.politics.misc
Comp and Rec	comp.graphics comp.sys.ibm.pc.hardware rec.motorcycles rec.sport.baseball
Sci and Rec	sci.crypt sci.med rec.autos rec.sport.baseball
Talk and Rec	talk.politics.mideast talk.politics.misc rec.autos rec.sport.baseball
Talk and Sci	talk.politics.misc talk.religion.misc sci.crypt sci.med

表 3-3 20Newsgroups 抽取的测试集上各个主题模型对应的分类器测试效果

Data set	PLSA	LDA	asymmetric LDA	TopicMapping	HLSM
Comp and Sci	0.761	0.771	0.792	0.831	0.855
Comp and Talk	0.785	0.790	0.813	0.846	0.871
Comp and Rec	0.770	0.776	0.781	0.834	0.853
Sci and Rec	0.724	0.723	0.767	0.803	0.822
Talk and Rec	0.811	0.802	0.832	0.821	0.876
Talk and Sci	0.804	0.811	0.839	0.847	0.867
Average	0.766	0.779	0.804	0.834	0.857

本分类问题，则由最有效的模型生成的文档主题概率向量可以比由其他模型生成的概率向量达到更好的性能。

为了测试 HLSM 的有效果，我们将其与以下几个具有代表性主题模型进行比较。

1. PLSA
2. symmetric LDA
3. asymmetric LDA
4. TopicMapping

在这个实验中，从 20Newsgroups 数据集中利用其原本具有的标记结构，生成了六个跨域文本数据集，每个数据集中有 4 个不同的类别领域，表3-2汇总了从 20Newsgroups 生成的数据集。考虑到如果是简单的二分类问题，主题模型抽取的主题向量的表达性，在该问题上可能无法得到良好的体现。为了使分类问题更有效和令人信服，实验中的文档分类任务被定义为多标签分类问题。在这些实验中，我们使用上述主题模型对每个数据集的所有文档估计概率 $p(t|d)$ ，并使用概率的向量 $p(t|d)$ 作为唯一的特征去训练用于多标签分类的支持向量机 (SVM) (modify)。对于每个数据集，随机的抽取 20% 的文档作为测试数据，其余 80% 标记的文档作为训练集训练 SVM 用于多标记分类。我们使用这些分类器来预测测试数据中未标记文档的类标签。注意，在每个数据集中有 4 个标记领域，只有当文档被分类到原始领域时，才认为分类的结果是正确的。

我们进行了与上一个实验相同的数据预处理。表3-3总结了每个数据集的所有主题模型基础上分类器的分类性能，前三列显示了 LDA，PLSA 和不对称 LDA 在不同数据集上调节主题数后能达到的最佳准确度。表的最后一行显示所有数据集的平均精度。从表中我们可以观察到依托于 HLSM 生成的主题概率向量训练的分类器在六个数据集上的准确度表现超过了其他所有的主题模型相应的分类器。

3.0.6 主题模型总结

(modify) 这一章中提出了一个全新的主题模型 HLSM，将社区检测领域的应用到主题生成的领域。我们将 HLSM 模型应用于文档建模和文档聚类的多个文档集合，并且针对现有技术方法的实验比较证明了该方法可靠的性能。特别地，主题中高 $p(word|topic)$ 的词的示例证明了 HLSM 可以在细微水平上区分主题。

我们的工作没有仅仅专注于标准主题模型算法的想法，它试图在解决方案空间中生成主题，并手动指定整个解空间的 *rank*。HLSM 通过揭示由语料库中的单词组成的网络的结构来生成主题。特别是，在社区检测领域的大量工作集中在随机块模型，试图揭示网络中的社区结构。我们认为这项工作与精神上的主题模式类似，将为主题建模提供新的想法和思路。

3.1 本章小结

在本章中本文先紧接上一章的内容讨论了复杂网络中社区发现和主题模型的共通之处，并紧接了提出了一种全新的主题建模思路 HLSM，即用社区发现的方法根据语料自动发现主题数 k 并生成一个靠近主题分布的先验，之后再优化这个先验。之后又详细的解释了这一建模方法的整个实现过程，并给出了详细的图解和公式推导，最后在主题模型领域公开的数据集 20newsgroup 上做了主题建模实验和文本分类实验对比了 HLSM 和其他经典主题模型间效果的对比验证了这一建模方法的有效性。并在最后深入分析实验结果，讨论这一方法和结合复杂网络分析这一思路的未来可研究点。

第四章 词嵌入 (word embedding) 算法和推荐系统的研究背景

4.1 推荐系统简介

4.1.1 推荐系统概念

推荐系统是信息过滤系统的子类，其试图预测用户对项目的偏好并产生有意义的建议。建议涉及各种决策过程，例如购买什么物品，听什么音乐，或什么在线新闻阅读^[13]。近年来推荐系统已经变得非常普遍，并且在各种领域中使用，包括电影，音乐，新闻，书籍，研究文章，搜索查询，社交标签和一般的产品等领域。推荐系统的设计不仅取决于应用的领域，还取决于可用数据源的特性。数据源可以包括用户与项目之间交互的数据，用户的特定属性或项目的特定属性，例如人口统计和产品说明等。推荐系统将采用不同的方式分析这些数据源来发掘用户和项目之间的关系，最终用于识别与用户匹配的项目^[14]。通常采用的推荐系统方法有两种——协同过滤方法，基于内容过滤或基于用户个性的方法^[15]。其中协作过滤系统仅分析历史相互作用的数据，而基于内容的过滤系统则是基于简档属性进行分析，混合技术试图组合两者的这些设计。目前，推荐系统的架构设计及用其评估现实世界的问题是一个活跃的研究领域。

4.1.2 近年研究状况

随着新兴的消费主义兴起，网络的出现使得买家正在被提供越来越大的范围的选择，而卖家面临着个性化广告的挑战。同时，企业也可以得到大量的事务数据，可以更深入地分析客户如何与产品供应空间进行交互。推荐系统的产生实现了基于数据分析自动生成推荐建议来满足买方和卖方的自然双重需求。

计算机能够提供建议的能力在计算历史上很早就被出现。Grundy 是一个机器的图书管理员，这是自动推荐系统的早期步骤。它的实现相当原始，基于简单的测试将用户分组为各个类型，再使用硬编码根据各种类型用户对书籍的偏好来生成建议。“协同过滤”是在第一个商业推荐系统 “Tapestry” 中被提出。“Tapestry” 是一个手动协同过滤系统：允许用户根据其他用户的 의견或行动在信息域中查询项目。之后又出现了自动协作过滤系统，GroupLens 使用这种技术识别特定用户可能感兴趣的

文章。协同过滤分析了用户之间使用的数据以找到匹配的用户-项目对，从而与在信息检索中的内容过滤方法并置。在此期间，推荐系统和协同过滤成为人机交互，机器学习和信息检索研究人员越来越感兴趣的话题。

早期的推荐系统的初始公式基于简单的相关统计和预测模型，不涉及对统计学和机器学习文献的实践。协同过滤问题被映射到分类，降维技术被用来提高解决方案的质量。同时人们试图将基于内容的方法与协作过滤相结合，并且将额外的领域知识并入推荐系统的体系结构中。之后，由于网络上公开了可用的数据集，以及电子商务额驱动，关于推荐系统的研究逐渐火热。

4.1.3 现有常见方法

推荐系统通常以两种方式产生推荐列表——协同过滤方法和基于内容的过滤或基于个性的方法^[16]。协同过滤方法是从用户的过去行为以及其他用户做出的类似决定构建模型，然后利用该模型预测用户可能感兴趣的项目^[14]。基于内容的过滤方法利用项目的一系列离散特征，以便推荐具有类似属性的附加项目^[17]。这些方法经常进行组合。

4.1.3.1 协同过滤

协同过滤被认为是推荐系统中最流行和广泛应用的技术^[18]。这种方法的基本实现原理是基于与该用户类似品味的其他用户过去喜欢的项目，向该用户推荐项目。其中两个用户的喜好相似性是基于用户的评价历史中的相似性来计算的。该方法背后的基本假设是，过去同意的人在未来也将同意，并且他们将喜欢类似与过去喜欢的项目，所以可以通过聚集其他用户的意见对活动用户的偏好的进行合理预测^[17]。

协同过滤方法可以分为两种类型：基于记忆的协同过滤和基于模型的协同过滤。基于记忆的协同过滤方法是使用用户的评级数据来计算用户或项目之间的相似性，这种方法也称为基于邻域的协同过滤方法^[18]。它的优点是结果可解释性强，容易创建和使用，容易促成新数据，项目的内容独立性好，适用于并发项目。它的缺点是当数据稀疏时性能降低，同时该表示方法依赖于特定的向量空间，这添加新项目变得复杂。基于模型的协同过滤方法通过估计建立的用户评级模型的参数来得到推荐建议。这种方法有一个更全面的目标，得到影响观察到的评级的潜在因素^[19]。大多数模型都是基于创建分类或者聚类技术来实现的。这种方法相比于基于记忆的协同过滤更适合稀疏的数据集，但是模型构建难度较大。

总体而言，协同过滤方法的优势在于其不依赖与项目的内容的分析，因此它可以精确地推荐复杂的项目，而不需要对项目本身进行理解。然而，协同过滤方法常常遇到的问题有以下三个，分别是冷启动，可伸缩性和稀疏性^[20]。

4.1.3.2 基于内容过滤

设计推荐系统时的另一种常见方法是基于内容的过滤。基于内容的过滤方法是基于项目的描述和用户偏好的简档实现的^[21]。该算法的基本原理是推荐类似于用户在过去喜欢的项目的项目。具体的过程是将各种候选项目与用户先前评价的项目进行比较，从而推荐最匹配的项目。

在基于内容的推荐系统中，需要利用特征描述项目，并且建立用户简档来指示该用户喜欢的项目类型。为了抽象系统中的项目的特征，需要应用项目表示算法。广泛使用的算法是 TF-IDF 表示算法（也称为向量空间表示）^[22]。而为了建立用户简档，需要关注关于用户的两种类型的信息：a. 描述用户喜好的模型；b. 用户与推荐系统交互的历史数据。

基于内容过滤的推荐系统对数据量要求低，但是它的局限性在于模型的迁移能力差，也就是说系统从用户在一个内容源的动作学习用户偏好，但是这种学习到的偏好应用于其他内容类型的适应性低。比如，基于用户已经浏览的新闻来为其推荐新闻文章是模型是有效的，但是当基于用户已经浏览的新闻来推荐音乐或产品时模型效果会不理想，但是这却是非常有意义的。

4.1.3.3 混合方法

为了利用基于内容过滤方法和协同过滤方法的优势，许多关于将这两者结合的混合方法被提出。混合方法的实现方式有以下几种：

1. 先分别利用基于内容过滤方法和协同过滤方法进行预测，之后将两个预测结果进行结合^[23]。Claypool 等人^[24] 使用自适应加权值来组合两个预测结果，其中协同过滤结果的权重随着访问项目的用户数量的增加而增大。
2. 将基于内容过滤的能力添加到基于协同过滤的方法中，反之亦然。Melville 等人^[25] 提出了内容增强协同过滤的一般框架，其中应用基于内容的预测将稀疏用户评级矩阵转换为完整评级矩阵，然后再使用协同过滤方法进行推荐预测。

3. 将基于内容过滤和协同过滤这两种方法统一为一个模型。Popescul 等人^[26]，同时考虑用户，项目和项目内容这三路数据。该生成模型假设用户选择潜在主题，并且从这些主题生成文档及其内容词。

几个研究成果比较了混合与纯协作和基于内容的方法的性能，并证明混合方法可以提供比纯方法更准确的建议。这些混合方法还可以用于克服推荐系统中的一些常见问题，例如冷启动和稀疏问题。

4.2 词嵌入（word embedding）学习算法介绍

词嵌入（word embedding）是自然语言处理领域（NLP）中一组关于语言建模和特征学习方向技术的集合名称。这项技术将来文档集中所有的单词或短语映射到一个在实数空间中的向量。数学概念上将，词嵌入（word embedding）是将每个单词从一个高维的 one-hot 的空间映射到低维的连续向量空间的方法。

简单来说，一个文档集中的所有单词构成一个单词表，那么我们能用一个 one-hot 向量直观表达这些单词，即我们构建一个 $1 \times N$ 的矩阵代表所有单词表中的词，在表达某一个单词的时候，我们让这个单词对应的那一维是 1，其他维为 0，则可以唯一的表达这个单词。但是这样我们的 one-hot 向量的长度等于词表大小会线性增长，显然其整个编码也是极为稀疏的，泛化性能和信息领用率都极低。如果我们能找到一个单词表对应的映射用一个连续的，低维的向量表达这些单词，那么在实际应用和表达性上都会有较大的提高。

词嵌入（word embedding）是目前成功应用无监督学习的少数几个应用之一。他们的主要好处可以说是他们不需要昂贵的标注，但可以从大量的未标准的语料库中获得。然后可以在使用少量标注数据的下游任务中使用预先训练的嵌入式表达（embedding）向量。

近年来生成此映射的方法包括神经网络^[27-29]，对单词共现矩阵的降维表达^[30-32]，以及在词出现的上下文中的显式表示^[33]。Bengio 等人的工作^[27]率先提出了用参数模型来学习单词的低纬空间连续表示。Collbert^[28]在之后提出了通过已有的语料集预先训练好单词的表示的想法，并从理论和实验上论证了这一思路的有效性，而 Mikolov 等人提出的 word2vec^[29]则是真正意义上让这一思路变得流行起来的工作，并且 word2vec 确实在很多实验上验证了其鲁棒的性能，之后大家试图从数学上解释 word2vec 的工作方式，Lebret 等人做了基于 PCA^[31] 的 word embedding 探索，

紧接着 Levy 等人^[30] 从矩阵分解的角度跟进研究 word embedding。本文在推荐系统中应用到的技术也是以 word2vec 为基础的方法，因此这里着重介绍一下 word2vec。

因为词嵌入 (word embedding) 是 NLP 的深度学习模型的关键构建块，所以通常假定 word2vec 也和深度徐埃领域的类似方法属于同一组技术。但是从技术上讲，word2vec 不被认为是深度学习的一部分，因为它的架构既不深，也不使用非线性（与 Bengio 的模型^[27] 和 Collobert 的模型^[28] 相反）。

Mikolov 等人的第一篇论文^[34] 中，提出了两种用于学习词嵌入的架构，它们比以前的模型在计算上复杂度更低。在他们的第二篇论文^[29] 中，他们改进这些模型，通过采用额外的策略，以提高训练的速度和准确性。这些架构和 Collobert 的模型和 Bengio 的语言模型相比有两个主要优点：1. 没有一味的加深网络的层数，2. 足够好的利用了词与词之间的上下文信息。

但是客观的评价，word2vec 的成功不仅仅依赖于上述的两个优点，其主要秘诀在于他们的训练策略，接下来将仔细介绍这些技巧。

4.2.0.1 连续词袋 (CBOW) 模型

因为语言模型是根据其预测语料库中的每个下一个词的能力来评估的，所以语言模型仅能够通过查看过去的词来预测下一个词，但是仅仅旨在生成准确的词嵌入的模型不受该限制的影响。因此使用目标词语 w_t 之前和之后的 $2n$ 个单词来预测它，如图4-1所示。他们称之为这个连续的词袋 (CBOW)，因为它使用连续的词表示，在这里顺序是不重要的。

模型的思路非常简单，用当前单词 w_t 前后窗口的词语去预估当前词 $p(w_t | w_{t-n}, \dots, w_{t+n})$ ，那么整个语料的概率就是：

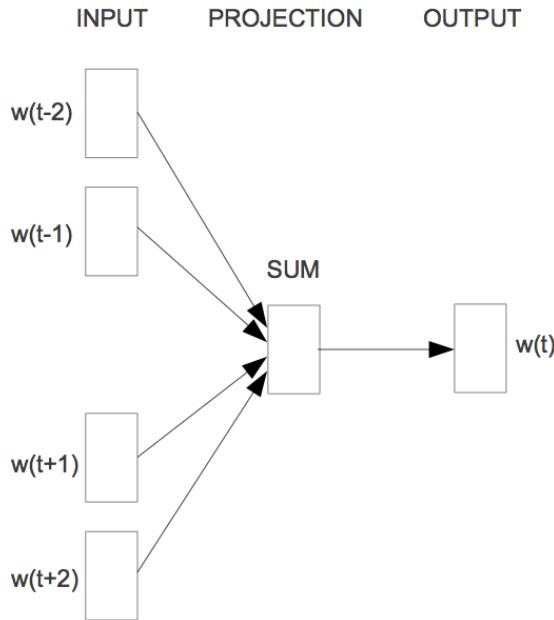
$$p(w_1, \dots, w_T) = \prod_i p(w_i | w_{i-n}, \dots, w_{i+n}) \quad (4-1)$$

同时可以计算整个模型的似然度为：

$$J_\theta = \frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}) \quad (4-2)$$

如果基于 n-gram 的思想，这个时候，计算一个单词的概率就是基于这个单词组成的 n-grams 的频率，即：

$$p(w_t | w_{t-n}, \dots, w_{t+n}) = \frac{\text{count}(w_{t-n}, \dots, w_{t+n})}{\text{count}(w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n})} \quad (4-3)$$

图 4-1 词袋模型^[34]

我们可以通过 softmax 层来计算数学等价的一个概率：

$$p(w_t | w_{t-n}, \dots, w_{t+n}) = \frac{\exp(h^\top v'_{w_t})}{\sum_{w_i \in V} \exp(h^\top v'_{w_i})} \quad (4-4)$$

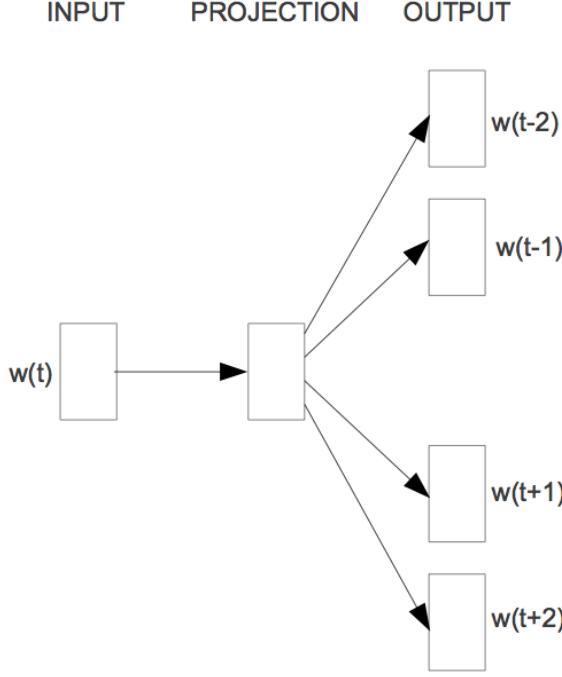
这里 V 和 h 都是我们需要学习的参数，每个单词 w 对应有两组参数，一组 v_w 是我们最终需要得到的词嵌入表示，另一组 h_w 是我们需要学习的隐层向量。CBOW 和 Bengio 提出的语言模型相比，唯一的改变就是 Bengio 的模型通过 w_t 之前的 n 个词来预测 w_t 而 CBOW 在每一个词 w_t 用其前后窗口 n 的词来做预测。

4.2.0.2 skip-gram 模型结构

虽然 CBOW 可以被看作是一种预认识语言模型，但是 skip-gram 却改变了语言模型的目标：skip-gram 使用中心词来预测周围的词，而不是像 CBOW 一样使用周围词预测中心词。如图4-2所示，skip-gram 的似然度是非常简单的将 w_t 前后的 n 个词的似然度相加：

$$J_\theta = \frac{1}{T} \sum_{t=1}^T \sum_{-n \leq j \leq n, j \neq 0} \log p(w_{t+j} | w_t) \quad (4-5)$$

同样的 softmax 被定义为：

图 4-2 skip-gram 模型^[34]

$$p(w_t | w_{t-1}, \dots, w_{t-n+1}) = \frac{\exp(h^\top v'_{w_t})}{\sum_{w_i \in V} \exp(h^\top v'_{w_i})} \quad (4-6)$$

不再去计算给出其先前词语后目标词 w_t 的概率，我们计算给定 w_t 的周围词语 w_{t+j} 的概率。因此，我们可以简单地在方程中替换这些变量：

$$(p(w_{t+j} | w_t) = \frac{\exp(h^\top v'_{w_{t+j}})}{\sum_{w_i \in V} \exp(h^\top v'_{w_i})} \quad (4-7)$$

同时 skip-gram 也不再如 CBOW 一样需要一个隐层的状态向量 h ，在这里 h 就是我们目标单词 w_t 的输入嵌入表示 v_{w_t} ，而我们最终得到的用于使用的目标单词 w_t 的输出向量定义为 v'_{w_t} ，将变量替换后：

$$p(w_{t+j} | w_t) = \frac{\exp(v_{w_t}^\top v'_{w_{t+j}})}{\sum_{w_i \in V} \exp(v_{w_t}^\top v'_{w_i})} \quad (4-8)$$

4.3 本章小结

此章节主要介绍了推荐系统和词嵌入算法的发展，介绍了现有的推荐系统的一些已有算法和简单介绍存在的一些问题。之后介绍了词嵌入的思想和发展过程并具体介绍了词嵌入中比较流行的 word2vec 算法。为后文讨论主题模型和词嵌入与推荐系统的创新结合做好铺垫。

第五章 主题模型和词嵌入（word embedding）与推荐系统的创新结合

本章会首先会讨论主题模型和词嵌入算法可以为推荐系统带来提高的空间，以及具体能解决哪些方面的问题，并在之后会详细的介绍其应用场景和应用方法。

5.1 主题模型和词嵌入（word embedding）在推荐系统中应用讨论

推荐系统发展到现在，已经开始在各个行业扮演着非常重要的角色，其地位不言自明。这个领域的技术可以说比较成熟了，因为从底层的数据流梳理到中层的算法应用以及最上层的实时工作引擎，都有了一套比较清晰的工作模式。但是也可以说不成熟，因为永远没有一个完美的系统，推荐系统总会有着各种各样需要解决的问题，如冷启动问题^[35]、同义的语义漂移问题^[36]、作弊攻击问题^[37]、长尾数据问题^[38]以及数据和计算规模问题。

本文中我们主要聚焦于主题模型和词嵌入算法在推荐系统中的应用，因此在这里我们主要讨论其可应用的场景以及能解决的问题，主题模型和词嵌入算法的最终目标可以看做是提炼出对于单词和文档的连续向量表示，那么这个表示能在推荐系统中解决的问题是冷启动、长尾数据问题、以及稀疏计算存储规模问题。

5.1.1 多场景的冷启动问题

在各个行业的推荐系统，其应用场景都是多样的，如商品推荐系统，在淘宝这样的大生态体系下，其需要推荐的场景非常之多。比如营销活动时的推荐，日常的猜你喜欢，购物链路上的推荐。这些场景是不同的，传统的推荐系统依赖的点击率预估技术，或者是协同过滤技术会有一个问题，他们依赖的样本量巨大。而一个新的场景开始运转时，其积累样本是需要时间的。但是如果我们将随机的为用户推送商品，这样积累数据的效率也非常低，因为用户往往看见的是自己不感兴趣的的商品，这样场景的用户流失率会增大以后用户就不再登录这个场景了。因此多场景的冷启动尤为关键，如何在初始时就尽快的积累有效的样本。

这时候主题模型和词嵌入算法的优势就得以体现出来。无论是主题模型还是词嵌入算法，其目标是根据已有数据得到对单词和文档有效的连续向量表达，这个向

量是面向于潜层语义的。在商品推荐领域，我们可以将每一个商品（item）看做是一个单词（word），一个用户（User）的行为链（即其有过行为的 item 列表）看做是文档（doc），这样就能简单的使用主题模型和词嵌入算法来对商品和用户进行向量表达。而主题模型和词嵌入算法都有一个特性，就是其抽取的向量表达的可迁移性即强泛化性。我们从一个已有的良好运转的场景中去收集用户（user）以及商品（item）的数据，从这些已有的数据中去通过主题模型和词嵌入算法推断商品和用户的向量表达，这些向量表达在新的场景中也是有一定效果的。对比传统的点击率预估技术，其工业界应用做法大量的使用稀疏的 ID 化逻辑回归模型，这样的模型的好处是其对高频样本的强拟合性，但是回到当前问题上，其模型一旦在一个固有的场景训练好，到了另一个场景由于数据分布的迁移，整个模型会几乎完全失效。因此主题模型和词嵌入算法合理的应用能有效的改善多场景的冷启动问题。

5.1.2 长尾数据问题

这个问题和上一小节中提到的冷启动问题有一点共通之处，就是其问题的本质是样本不够充足。不同之处在于，长尾数据问题中是有一部分的样本非常充足，而低频样本的贫乏导致了这些样本所代表的真实情形往往在推荐系统中被估计错误或偏移，而推荐系统又是一个非常强的自反馈系统有强马太效应，导致了推荐系统对那些低频数据处理得越来越差，一直无法让长尾数据得到有效的解决。更有意思的是，在商品推荐领域中长尾数据的占比还非常大，其分布也更为复杂，如图5-1。

长尾分布是常见的，分布本身不会造成问题，造成问题的是常见的推荐系模式。如图5-2，常见的推荐系统的样本分布空间分为三部分，训练样本空间、在线需要推断的样本空间以及全量样本空间。如果我们将一个用户（user）和一个商品（item）组成的对称为一个样本，训练样本空间即我们可以观测到的有反馈行为（点击与否，购买与否，收藏与否等等）的样本对，这个量是由推荐系统所服务的场景的用户使用情况有关的，同时全量样本空间是非常大的其数量等于用户数乘以商品数，远远大于训练样本空间，因为不可能所有的用户都对所有的商品做出过反馈。而在在线需要推断的样本空间指的是在线系统中一个用户发送了请求之后，推荐系统需要给出的推荐分数，或者排序关系的样本对即给每个用户最终结果之前的商品候选集和用户间组成的样本。值得注意的是，这个数量也一定比训练样本空间大，因为推荐系统往往是在全量样本空间中通过一些 `match` 的方法，缩小了候选集之后，再用更精细的模型去进行筛选。而最终通过筛选出的商品被用户看到，并作出反馈行为，成

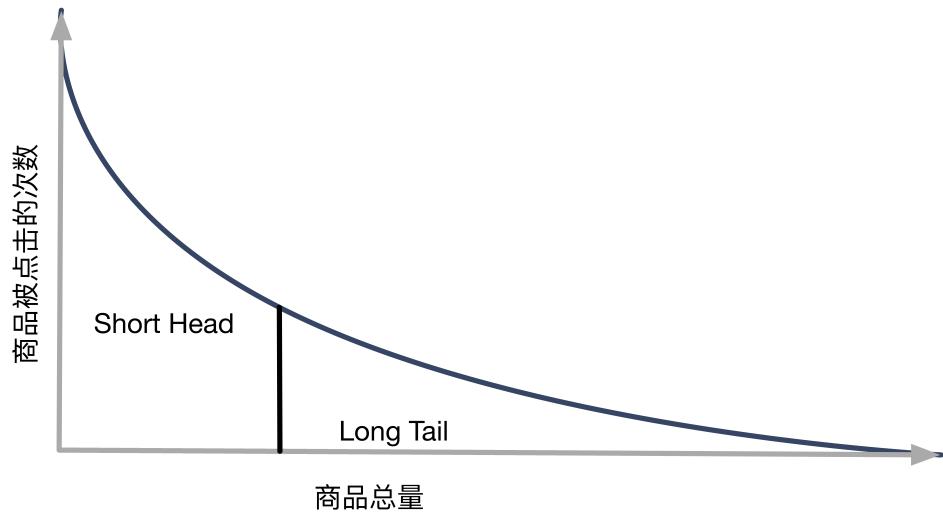


图 5-1 商品推荐中的长尾效应

为系统可以收集的训练样本，未被用户看到的样本推荐系统其实无法定义这些样本的性质（label）。

清晰了推荐系统的样本分布之后，其实长尾问题带来的影响就明确了，那些活跃的商品，会被系统推送出来更多的次数，与此同时这些活跃的商品就更多的进入了训练样本空间，推荐系统对这些商品的学习变得更精准。反之那些低频的商品就更少的被展现，推荐系统也就更难收集到这些商品的样本进行学习，对这些商品的评价就越来越不精准。这样的一个反馈系统，会让用户总是看到一些热门的商品，这会极大的损伤用户的体验。

而如上一小节提到，主题模型和词嵌入学习算法通过合理的设计能具有较强的泛化性。并且这两种算法思路在定义模型的时候，不会以商品和用户组成的对为单位来定义样本，其样本是行为列表，这个不同会使得这些方法更好的学习到每个商品和每个用户的潜层语义，因此值得尝试用主题模型和词嵌入学习的思路来解决长尾问题。

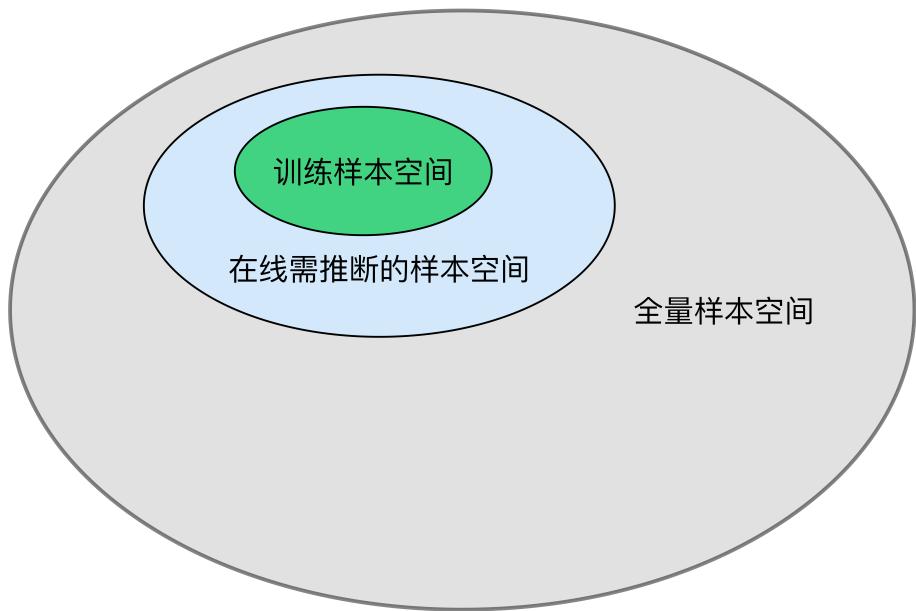


图 5-2 推荐系统的样本分布

5.1.3 稀疏数据的计算存储规模问题

推荐系统抽象出来的问题是对一个用户推荐一些商品，或者将一个商品推荐给可能感兴趣的用户群体，而系统中用于表达用户的特征是用户对各种商品的行为，对商品的表达是在这个商品上有过各种行为的用户列表。这样的数据是由次数和代表用户或者商品的 ID 组成的，是非常宽维度并且稀疏的，实际应用中往往使用哈希的方法来存储，但是其开销的空间是巨大的。而且真实的在线系统中虽然哈希的方法在取一个 ID 对应值的复杂度可以认为是 $O(1)$ ，但是这个开销也和用户的行为次数呈线性相关。

如果现在每个用户和每个商品都用一个固定维度的连续向量表达，这样对系统无论从计算还是存储方面带来的便利性都是极大的。

5.2 主题模型在推荐系统中的结合

如前文中提到，将每个用户（user）的行为列表（item list）看做是文档（doc），每个商品（item）定义为单词（word）就可以很简单的用主题模型对之建模。这节会

详细的介绍在商品推荐问题上，如何对用户和商品进行建模，并得到它们的潜层概率空间语义表达。

主题建模的方法有很多，包括之前介绍的 PLSA、LDA、非对称 LDA 以及我们在这篇论文中创新提出的 HLSM。但是在真实的商品推荐场景下，对应于文档 (doc) 的用户量是亿级别的，对应于单词 (word) 的商品量也是亿级别的。这样的数据级别上，并行计算的 PLSA 是比较好的选择，而本文提出的 HLSM 由于其中需要对单词和文档共现矩阵进行非负矩阵分解，目前没有很好的资源能解决这一问题，同时 LDA 虽然有其并行计算版本 PLDA^[39] 和 MSRA 提出的 Light LDA^[40]，但受资源限制选择设计了以 PLSA 为基础的方法。

和 PLSA 一样，这里我们要估计的是两个概率分布 $p(topic|user)$ (后文中简写为 $p(t|u)$) 与 $p(item|topic)$ (后文简写为 $p(i|t)$)。则整个训练数据集的概率 $p(item, user)$ 为：

$$P(\mathbf{item}, \mathbf{user}) = \prod_{s=1}^N \prod_{j=1}^M P(u_s, i_j)^{n(u_s, i_j)} \quad (5-1)$$

则似然度函数为：

$$\begin{aligned} \log P(\mathbf{u}, \mathbf{i}) &= \sum_{s=1}^N \sum_{j=1}^M n(u_s, i_j) \log P(u_s, i_j) \\ &= \sum_{s=1}^N n(u_s) [\log P(u_s) + \sum_{j=1}^M \frac{n(u_s, i_j)}{n(u_s)} \log \sum_{k=1}^K P(i_j | t_k) P(t_k | u_s)] \\ &\propto \sum_{s=1}^N \sum_{j=1}^M n(u_s, i_j) \log \sum_{k=1}^K P(i_j | t_k) P(t_k | u_s) \end{aligned} \quad (5-2)$$

用 EM 算法来进行推断，则对应的 Q 方法函数为：

$$\begin{aligned}
 Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(i)}) &= \sum_{t_1, \dots, t_n} \sum_{l=1}^n \log P(\mathbf{u}, \mathbf{i}, t_l) P(t_1, \dots, t_n | \mathbf{u}, \mathbf{i}) \\
 &= \sum_{t_1, t_2, \dots, t_n} [\log P(\mathbf{u}, \mathbf{i}, t_1) + \dots + \log P(\mathbf{u}, \mathbf{i}, t_n)] \\
 &\quad P(t_1 | \mathbf{u}, \mathbf{i}) P(t_2 | \mathbf{u}, \mathbf{i}) \dots P(t_n | \mathbf{u}) \\
 &= \sum_{t_1, t_2, \dots, t_n} [\log P(u_1, i_1, t_1) + \dots + \log P(u_n, i_n, t_n)] \\
 &\quad P(t_1 | u_1, i_1) P(t_2 | u_1, i_1) \dots P(t_n | u_1, i_1) \\
 &= \sum_{s=1}^N \sum_{j=1}^M n(u_s, i_j) \sum_{k=1}^K P(t_k | u_s, i_j) \log P(i_j | t_k) P(t_k | u_s)
 \end{aligned} \tag{5-3}$$

由此 E step 为：

$$P(t_k | u_s, i_j) = \frac{P(i_j | t_k) P(t_k | u_s)}{\sum_{l=1}^K P(i_j | t_l) P(t_l | u_s)} \tag{5-4}$$

M step 为：

$$P(i_j | t_k) = \frac{\sum_{s=1}^N n(u_s, i_j) P(t_k | u_s, i_j)}{\sum_{m=1}^M \sum_{s=1}^N n(u_s, i_m) P(t_k | u_s, i_m)} \tag{5-5}$$

$$P(t_k | u_s) = \frac{\sum_{j=1}^M n(u_s, i_j) P(t_k | u_s, i_j)}{n(u_s)} \tag{5-6}$$

其中 $n(u_s)$ 表达的是用户 i 的行为次数。

而在真实的使用场景中，用户的行为列表会发生非常快的变化，如一个淘宝用户会不时的进行商品点击、购买、收藏等行为，其行为列表的更新频率非常的高。如果我们用上述公式中的 $p(t|u)$ 来表示用户，这个概率分布是过去用户的潜层语义空间的表示，而不是当下用户实时的语义表示。而更为稳定的表达是对于商品的，因为商品本身的属性随时间迁移的变化并不会特别明显，至少在天级别的更替中，我们认为商品的属性不会发生太大的变化。所以在真实的使用过程中，我们仅仅固定了一个概率分布 $p(t|i) = \frac{p(t) \cdot p(i|t)}{p(i)}$ 为商品的向量表达 $Item_{vec}$ ，其中 $p(t) = \frac{\sum_{s=1}^N p(t|u_s)}{N}$ 。将用户的向量表达 $User_{vec}$ 定义为其所有行为列表中商品对应的 $p(i|t)$ 的加权平均：

$$User_{vec} = \frac{\sum_{j=1}^M n(u, i_j) \overrightarrow{p(i|t)}}{n(u)} \tag{5-7}$$

在获取到 $User_{vec}$ 和 $Item_{vec}$ 之后，我们可以定义一个函数 $f(User_{vec}, Item_{vec})$ 来计算 User 和 Item 的相关性分数：

$$Score_{ui} = f(User_{vec}^u, Item_{vec}^i) \quad (5-8)$$

最直观的，我们可以将之定义为余弦距离：

$$Score_{ui} = \cos User_{vec}^u \cdot Item_{vec}^i \quad (5-9)$$

当然，我们做了更复杂一点的尝试，用一个深度神经网络来学习 f ，如图5-3。这里可以用已经观测过有反馈的 User 和 Item 组成的 pair 样本来训练，如有点击 label 是 1 没有点击 label 为 0。这么做的动机是因为 $User_{vec}$ 和 $Item_{vec}$ 的关系函数并不是已知的，而深度神经网络的拟合能力非常强，在有足够的样本的情况下，其可以很好的拟合出 $User_{vec}$ 和 $Item_{vec}$ 的函数关系。不过缺点是其计算与训练的开销都不如之前提到的计算余弦距离那么直接。在之后的实验章节中会详细介绍这两种方法间的对比效果。

综上所述，主题模型在推荐系统中应用的整个步骤为：

步骤 1. 将用户 (User) 的行为数据整理为商品链 (Item list) 作为文档，每个商品作为单词

步骤 2. 在上一步产出的数据中用 PLSA 通过 EM 算法去迭代推断 $p(i, t)$

步骤 3. 计算 $p(t) = \frac{\sum_{s=1}^N p(t|u_s)}{N}$ 和 $p(t|i) = \frac{p(t) \cdot p(i|t)}{p(i)}$

步骤 4. 存储 $\overrightarrow{p}(t|i)$ 为 $Item_{vec}$

步骤 5. 计算 $User_{vec} = \frac{\sum_{j=1}^M n(u, i_j) \overrightarrow{p(i|t)}}{n(u)}$

步骤 6. 在线对于 $Item_i$ 和 $User_u$ 通过定义好的函数 $f(User_{vec}, Item_{vec})$ 计算 $Score_{ui} = \cos User_{vec}^u \cdot Item_{vec}^i$

本文在之后的实验章节中也会详细介绍在实际应用场景中对此种方法的尝试的环境以及最终的实验结果，由此来验证这一想法的有效性。

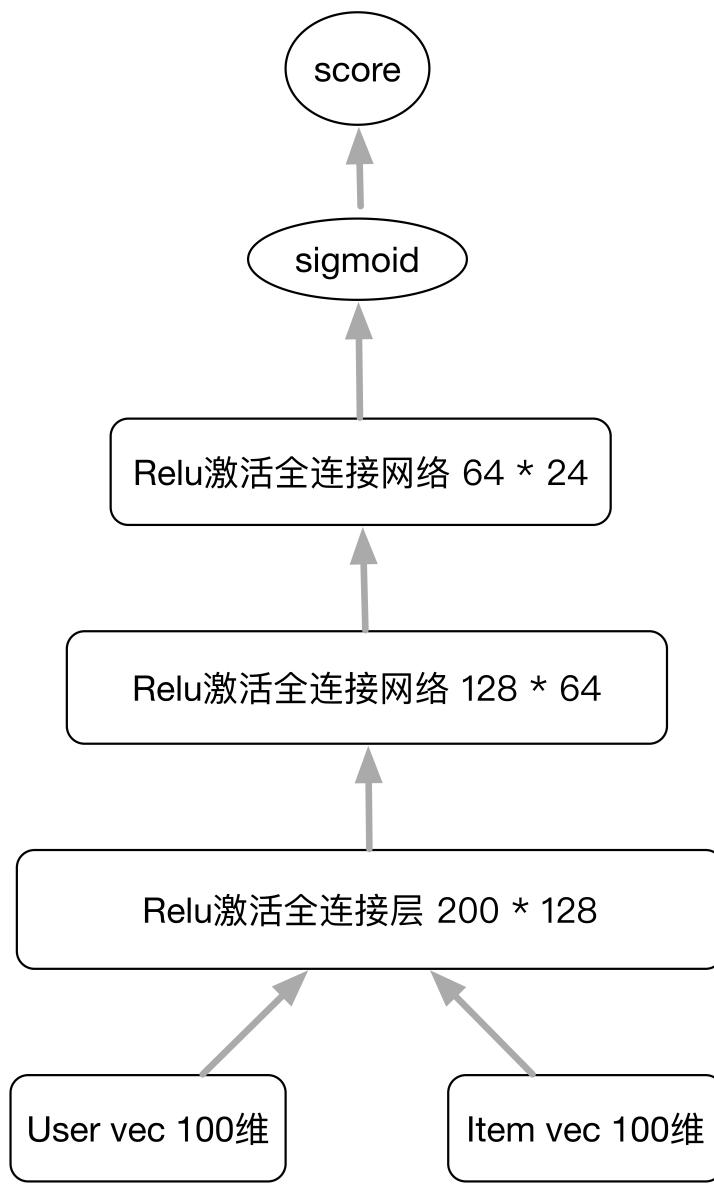


图 5.3 User 和 Item 关系学习的全连接神经网络

5.3 类 word2vec 词嵌入方法在推荐系统中的结合

这一小节将主要介绍类似 word2vec 这种和最终使用场景不直接相关的词嵌入方法在推荐系统中的结合，在下一小节将介绍本文提出的一种在推荐系统中和使用场

景相关性稍强的弱监督词嵌入方法。

5.3.1 基于用户行为序列的 word2vec

和前文分析一样，将用户在商品上的行为序列抽象为文档，则单词就是每一件商品。这里直观的去思考用户行为序列的生成过程，为了更好的去理解，本文摘取了一个实际的用户行为序列（已经人工的去掉了所有图例中的商标）用于展示。

如图5-4，可以发现用户的行为是有明显的局部连续一致性的，如果考虑到周围的商品之间的关系，可以看到在一定区域内商品相邻的商品是相似的。这个可以很直观的理解，一个用户在商城浏览的时候一定会在某一兴趣点上连续的浏览，直到选中自己心仪的商品或者被其他兴趣点相关的商品吸引开。此图中的最左端商品就可以看到，用户的兴趣从女鞋到了女裤。这里女鞋和女裤可以是搭配关系，也算有一定的相似性，如果简单的用 n-gram 信息进行学习也是能有收益的。可能拟合出具有这种搭配关系的兴趣迁移。

但是如图5-5，用户的兴趣可能是跳跃的，男裤之后是零食之后又到女鞋，这一行为的捕捉是非常困难的，这样的行为序列占比还非常大，而相邻的商品彼此之间可能出现兴趣的断点，如果在这里直接用 n-gram 去监督我们的 word2vec 显然是不合理的，因为在零食后面出现任何兴趣点都有可能，而这完全是噪声。

为了避免这些噪声，需要先研究清楚这种噪声的分布情况。根据消费者进入网络商城的目标是要选择或者只是观察自己想要的商品，而兴趣的巨大跳跃是和时间有关系的，人在短时间之内去大尺度的改变一个兴趣点的可能会比较小。但是直观的理解显然是不严谨的，因此本文为了探索时间和兴趣点迁移的关系做了严谨的数据统计。

统计的目标很简单，本文在这里认为跨类目粗略算作是兴趣点的迁移，统计两次行为之间相邻时间长短 ΔT 和两次行为商品属于不同类目的比例 δ ，对用户的行为做抽样统计了两者间的关系，如图5-6。

从图中我们可以看到两次行为的商品跨类目比例 δ 与两次行为的时间间隔 ΔT 成递增关系并且是一个类对数分布。这说明之前的直观理解和实际的数据分布是相符合的。根据这个数据分布，本文简单的用一个随时间间隔 ΔT 递增的概率函数：

$$g(\Delta T) = \frac{1}{1 + e^{-(\log_{10}^{\Delta T} + a + b)}} \quad (5-10)$$

来处理用户的行为序列，其中 a 和 b 是可以调控的超参。具体来说就是，在一个完

整的用户行为序列中，遍历其每一个行为，在每一个行为时刻根据距离上一个行为的时间间隔以 $g(\Delta T)$ 的概率采样选择是否将行为序列切断，切断后的行为序列才当做一个文档用 word2vec 去训练每个商品为单词的语义向量 $Item_{vec}$ 。

为什么不直接在两次连续行为类目不同时就切断行为序列成为不同的两份文档，而要这么稍显复杂的进行采样呢？因为前文中提到跨类目间也可能是有联系的，很多时候这代表一种搭配关系，如用户看了手机壳之后可能会想看看手机膜，如果完全按照类目将用户的行为序列切断，则后面的 word2vec 算法接收的输入会完全丧失学习商品间搭配关系的能力。本文在这里使用一个和时间间隔 ΔT 相关的采样方法来切断用户的行为虽然可能引入一部分噪声，但是让后续学习到的商品词向量 $Item_{vec}$ 能表示的语义里可能蕴含搭配以及用户天然跨类目兴趣。

在获取到所有的商品词向量 $Item_{vec}$ 表达后，和上一节文中提到的计算用户向量表达 $User_{vec}$ 一样，本文对用户有过行为的所有商品，依据用户对这些商品的行为次数，进行加权平均得到用户的向量表达：

$$User_{vec} = \frac{\sum_{j=1}^M n(u, i_j) \cdot Item_{vec}}{n(u)} \quad (5-11)$$

之后的做法也和上一节中一样，计算用户和商品的关联分数可以直接用余弦距离或者是用当下场景积累的样本来直接用神经网络拟合，具体就不再赘述。

最后总结，基于用户行为序列的 word2vec 在推荐系统中应用的整个步骤为：

步骤 1. 遍历每个用户的行为序列，根据连续行为直接的时间间隔 ΔT 用 $g(\Delta T) = \frac{1}{1+e^{-(\log_{10}^{\Delta T} + a + b)}}$ 对其进行概率采样决定是否分裂为不同的两份文档。

步骤 2. 用上一步生成的文档训练 word2vec，为了效率具体采用 skip-gram 的负采样版本，获得商品的向量表示 $Item_{vec}$

步骤 3. 计算 $User_{vec} = \frac{\sum_{j=1}^M n(u, i_j) \cdot Item_{vec}}{n(u)}$

步骤 4. 在线对于 $Item_i$ 和 $User_u$ 通过定义好的函数 $f(User_{vec}, Item_{vec})$ 计算 $Score_{ui} = \cos User_{vec}^u \cdot Item_{vec}^i$

后面的实验章节会具体介绍这一种方法的应用场景与具体效果。

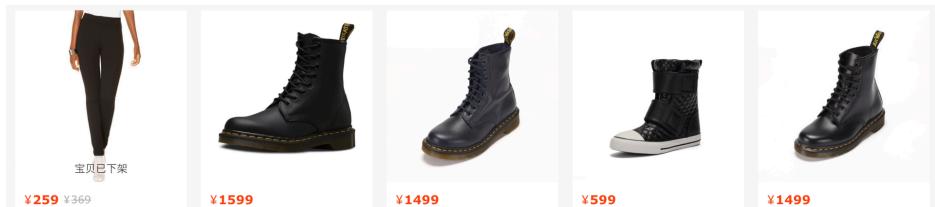


图 5-4 网络商城中用户的行为足迹 1



图 5-5 网络商城中用户的行为足迹 2

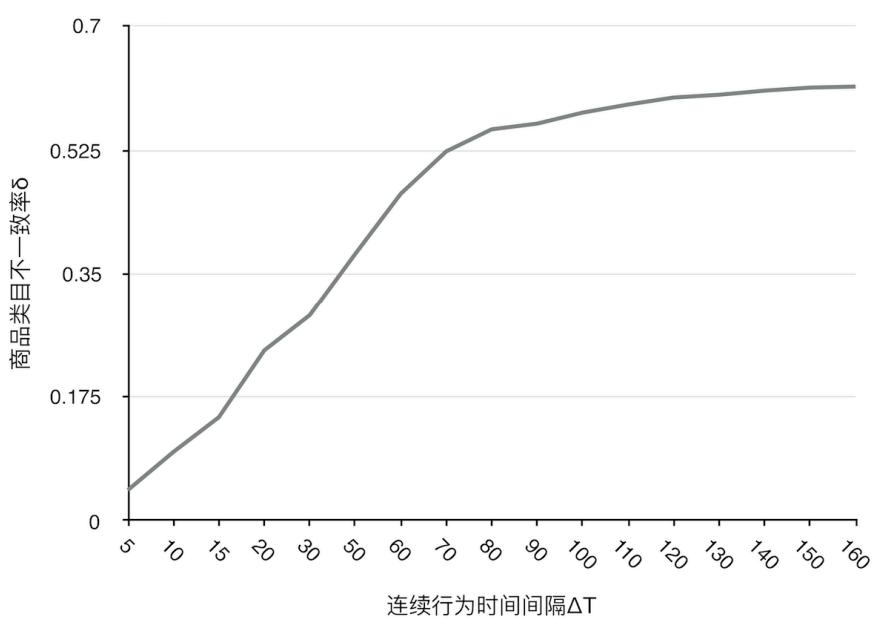


图 5-6 两次连续行为跨类目比例和时间间隔的关系分布

5.3.2 基于商品相似度图生成商品词向量

上一小节中提到基于用户的行为序列生成商品词向量，这是一种最简单直接的思想，目标是通过真实的用户行为，去学习所有的商品和用户的向量表达。但是本文还提出另一种思路基于商品相似度来学习商品的向量表达。

这一思路主要借鉴于 node2vec^[41]一文中提出的在一个有权图中，基于随机游走来学习整个网络的思路。在商品推荐的应用场景下，很多时候不仅仅是要给用户推荐根据他过去所有行为的兴趣点的商品，有时候需要固定某一个兴趣点来推荐，比如找相似商品之类的场景。这种时候虽然也希望发散用户的兴趣点，但是又希望约束能和某一兴趣的有足够联系。因此想到用所有商品作为节点（node），其相似度作为权重构建一个网络，之后以随机游走的方式将随机游走过程中经过的路径作为文档用 word2vec 进行学习。这种方式生成的向量表达会更倾向于让相似的商品的在向量空间距离更近。

这里对于计算商品相似度的算法没有特殊的要求，简单的采用了传统的协同过滤（modify）。下面介绍一下随机游走的过程。在一个已经构建好的网络 E 中，我们已经知道了所有的节点（node）和其邻居（neighbor）之间所有边的权重 w ，随机游走最关键需要知道的就是下一次游走代理需要去到哪一个节点，即下一步到所有邻居节点的概率。最原始的方法，在行走过程中，第 $i-1$ 步的节点 n_{i-1} 为 u ，其到达下一个节点 n_i 为 v 的概率为：

$$P(n_i = v | n_{i-1} = u) = \begin{cases} \frac{\pi_{uv}}{C} & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \quad (5-12)$$

其中 π_{uv} 是 u 跳转到 v 未归一化的概率，而 C 是归一化常量。最简单的本文 π_{uv} 直接等于 u 和 v 之间的权重 w_{uv} 。

5.4 主题模型和词嵌入在推荐系统中应用实验

参考文献

- [1] Barathi B. Cross-domain text classification using semantic based approach[A]. // Sustainable Energy and Intelligent Systems (SEISCON 2011), International Conference on[C]. 2011: 820–825.
- [2] Chang H C, Hsu C C. Using topic keyword clusters for automatic document clustering[A]. // Information Technology and Applications, 2005. ICITA 2005. Third International Conference on[C]. 2005: 419–424 vol.1.
- [3] Hofmann T. Probabilistic latent semantic indexing[J]. 1999: 50–57.

- [4] Blei A Y D M; Ng, Jordan M I. Latent dirichlet allocation[J]. 2003: 993–1022.
- [5] Sontag D, Roy D. Complexity of Inference in Latent Dirichlet Allocation[A]. // Advances in Neural Information Processing Systems 24[C]. Curran Associates, Inc., 2011: 1008–1016.
- [6] Blei D, Carin L, Dunson D. Probabilistic Topic Models[J]. 2010, 27(6): 55–65.
- [7] Wallach H M, Mimno D M, McCallum A. Rethinking LDA: Why Priors Matter[A]. // Advances in Neural Information Processing Systems 22[C]. Curran Associates, Inc., 2009: 1973–1981.
- [8] Rosvall M, Bergstrom C T. Maps of random walks on complex networks reveal community structure[J]. 2008, 105(4): 1118–1123.
- [9] Karrer B, Newman M E J. Stochastic blockmodels and community structure in networks[J]. 2010, abs/1008.3926.
- [10] Peixoto T P. Hierarchical block structures and high-resolution model selection in large networks.[J]. 2013, 4(1).
- [11] Larremore D B, Clauset A, Jacobs A Z. Efficiently inferring community structure in bipartite networks.[J]. 2014, abs/1403.2933.
- [12] Rosvall M, Bergstrom C T. Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems[J]. 2011, 6(4): e18209.
- [13] .
- [14] Encyclopedia of Machine Learning[M]. Springer, 2010.
- [15] Doucet A, Ahonen-Myka H. Naïve clustering of a large XML document collection[A]. // INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the First INEX Workshop. Dagstuhl, Germany, December 8-11, 2002[C]. Sophia Antipolis, France: ERCIM, 2002: 81–87.
- [16] Mooney R J, Roy L. Content-based book recommending using learning for text categorization.[A]. // ACM DL[C]. 2000: 195–204.
- [17] Herlocker J, Konstan J, Terveen L, et al. Evaluating collaborative filtering recommender systems[J]. 2004, 22(1): 5–53.
- [18] Breese J S, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[A]. 1998.
- [19] Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering[J]. 2010, 4(1).
- [20] Lee S, Yang J, Park S Y. Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem.[A]. // Discovery Science[C]. Springer, 2004: 396–402.
- [21] Brusilovsky P, Maybury M. From Adaptive Hypermedia to the Adaptive Web[J]. 2002, 45(5): 31–33.
- [22] Rajaraman A, Leskovec J, Ullman J D. 2014.
- [23] Smyth B, Cotter P. PTV: Intelligent Personalised TV Guides.[A]. // Proceedings of the 12th Conference on Innovative Applications of Artificial Intelligence. (IAAI-2000)[C]. AAAI Press, 2000.
- [24] Claypool M, Gokhale A, Miranda T, et al. Combining content-based and collaborative filters in an online newspaper[A]. // Proceedings of ACM SIGIR Workshop on Recommender Systems[C]. 1999.
- [25] Melville P, Mooney R J, Nagarajan R. Content-Boosted Collaborative Filtering for Improved Recommendations.[A]. // AAAI/IAAI[C]. AAAI Press / The MIT Press, 2002: 187–192.

- [26] Popescul A, Ungar L, Pennock D, et al. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments[A]. // Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence[C]. 2001: 437–444.
- [27] Bengio Y, Ducharme R, Vincent P. A Neural Probabilistic Language Model.[A]. // NIPS[C]. MIT Press, 2000: 932–938.
- [28] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning.[A]. // ICML[C]. ACM, 2008: 160–167.
- [29] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[A]. // Advances in Neural Information Processing Systems 26[C]. Curran Associates, Inc., 2013: 3111–3119.
- [30] Levy O, Goldberg Y. Neural Word Embedding as Implicit Matrix Factorization[A]. // Advances in Neural Information Processing Systems 27[C]. Curran Associates, Inc., 2014: 2177–2185.
- [31] Lebret R, Lebret R. Word Emddeddings through Hellinger PCA.[J]. 2013, abs/1312.5542.
- [32] .
- [33] Levy O, Goldberg Y, Ramat-Gan I. Linguistic Regularities in Sparse and Explicit Word Representations.[A]. // CoNLL[C]. 2014: 171–180.
- [34] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. 2013, abs/1301.3781.
- [35] Schafer J B, Frankowski d, Herlocker J, et al. Collaborative Filtering Recommender Systems[A]. // The Adaptive Web[C]. Springer, 2007.
- [36] Su X, Khoshgoftaar T. A survey of collaborative filtering techniques[J]. 2009, 2009: 4.
- [37] Zhang F. A Survey of Shilling Attacks in Collaborative Filtering Recommender Systems[A]. // Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on[C]. 2009: 1–4.
- [38] Applied Probability and Queues[M]. New York, NY: Springer New York, 2003: 266–301.
- [39] Wang Y, Bai H, Stanton M, et al. PLDA: Parallel Latent Dirichlet Allocation for Large-Scale Applications.[A]. // AAIM[C]. Springer, 2009: 301–314.
- [40] Yuan J, Gao F, Ho Q, et al. LightLDA: Big Topic Models on Modest Computer Clusters.[A]. // WWW[C]. ACM, 2015: 1351–1361.
- [41] Grover A, Leskovec J. node2vec: Scalable Feature Learning for Networks[A]. 2016. cite arxiv:1607.00653Comment: In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

附录 A 不定型 (0/0) 极限的计算

定理 A.1 (L'Hospital 法则) 若

1. 当 $x \rightarrow a$ 时, 函数 $f(x)$ 和 $g(x)$ 都趋于零;
2. 在点 a 某去心邻域内, $f'(x)$ 和 $g'(x)$ 都存在, 且 $g'(x) \neq 0$;
3. $\lim_{x \rightarrow a} \frac{f'(x)}{g'(x)}$ 存在 (或为无穷大),

那么

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)}. \quad (\text{A-1})$$

证明: 以下只证明两函数 $f(x)$ 和 $g(x)$ 在 $x = a$ 为光滑函数的情形。由于 $f(a) = g(a) = 0$, 原极限可以重写为

$$\lim_{x \rightarrow a} \frac{f(x) - f(a)}{g(x) - g(a)}.$$

对分子分母同时除以 $(x - a)$, 得到

$$\lim_{x \rightarrow a} \frac{\frac{f(x) - f(a)}{x - a}}{\frac{g(x) - g(a)}{x - a}} = \frac{\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}}{\lim_{x \rightarrow a} \frac{g(x) - g(a)}{x - a}}.$$

分子分母各得一差商极限, 即函数 $f(x)$ 和 $g(x)$ 分别在 $x = a$ 处的导数

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{f'(a)}{g'(a)}.$$

由光滑函数的导函数必为一光滑函数, 故 (A-1) 得证。 \square

致 谢

感谢 Donald Ervin Knuth.

攻读学位期间发表的学术论文目录

期刊论文

- [1] **Zhang San** , Newton I, Hawking S W, et al. An extended brief history of time[J]. Journal of Galaxy, 2079, 1234(4): 567–890. (SCI 收录, 检索号: 786FZ) .

会议论文

- [1] McClane J, McClane L, Gennero H, et al. Transcript in Die hard[A]. // Proc. HDDD 100th Super Technology Conference (STC 2046)[C]. Eta Cygni, Cygnus: 2046: 123–456. (EI 源刊) .

专利

- [1] 张三 , 李四. 一种进行时空旅行的装置 [P]. 中国: 1234567, 2046-01-09.