



南京大學

研究生畢業論文 (申請碩士學位)

論文題目 基于聚类信息结合的视频推荐与缓存技术研究

作者姓名 林涛

学科、专业方向 计算机技术

指导教师 陆桑璐 教授 叶保留 教授

研究方向 分布式计算与并行处理

2016 年 5 月

学 号 : MF1333025

论文答辩日期 : 2016 年 6 月 1 日

指 导 教 师 : (签字)

Research On Clustering Incorporation Based Video Recommendation and Caching

by
Tao Lin

Directed by
Professor Sanglu Lu, Professor BaoLiu Ye

Department of Computer Science and Technology
Nanjing University

May 2016

*Submitted in partial fulfilment of the requirements
for the degree of Master in Computer Technology*

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目： 基于聚类信息结合的视频推荐与缓存技术研究
计算机技术 专业 2013 级硕士生姓名： 林涛
指导教师（姓名、职称）： 陆桑璐 教授 叶保留 教授

摘 要

关键词：

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research On Clustering Incorporation Based Video Recommendation and Caching

SPECIALIZATION: Computer Technology

POSTGRADUATE: Tao Lin

MENTOR: Professor Sanglu Lu, Professor BaoLiu Ye

Abstract

Keywords:

Contents

List of Tables

List of Figures

第一章 绪论

1.1 引言

过去几年，蜂窝网络技术持续着迅猛增长的态势。尤其在数据通信方面，已经保持着每年一个数量级的增长，早已经把语音通信远远甩在后面。到2016年，全球移动数据通信预期将增长到每个月10.8 $exabytes$ ($1exa = 10^{18}$),是2011年的18倍[?]。这一不可预期的增长正在对现存的蜂窝网络提出新的挑战。移动数据卸载是应对这一挑战提出的新理念，它指利用现存的互补网络技术以及创新的数据传输技术，来缓解在蜂窝网络中的拥塞以及使得网络资源被更好地利用。数据卸载的目标是维持用户在网络中的服务质量，同时减少数据传输饥饿带来的影响。随着数据通信在移动网络中的持续增长，数据卸载将在不久的将来成为蜂窝网络中关键的组成部分。

数据卸载的最初驱动因素是数据通信的兴起。数据通信的正在对蜂窝网络中产生更多的拥塞并最终降低用户的体验。这一兴起归咎于如下因素：

1. 逐渐普及的高性能移动设备，如移动手机，平板电脑等。这些设备将产生是基本功能的移动手机将近35倍的数据量。
2. 每个设备的通信量的增长。由于数据通信速度的提升以及移动设备电池续航量的提升，每个设备将占用信道更多的时间数量占比。
3. 移动视频内容数量的增长。移动视频比其他的网络内容有更高的码率，加之设备不断扩大的屏幕都将需要更多的数据传输。
4. 随着移动宽带的引入。移动宽带的定价已经可以与固定宽带的定价比拟，加之日益流行的普适移动计算都是用户数量不断增长。

无线网络视频传输已经成为无线通信进步的主要驱动力，而且其影响力还在继续增长。无线网络中的视频一般指的是简短的视频片段（例如YouTube或者新闻频道），这些视频一般在小型移动设备设备上显示。移动设备的不断更新换代，性能不断的提升，使得在移动设备上观看视频已经成为生活的常态。增长的用户群，不断提升的分辨率已经越来越大的设备屏幕大大增加了无线传输的负载，导致视频传输占据着无线通信中的主要负载类型。近期的工业界估计视频传输很快就要超越普通的互联网内容传输，并且将网络流量提升两个数量

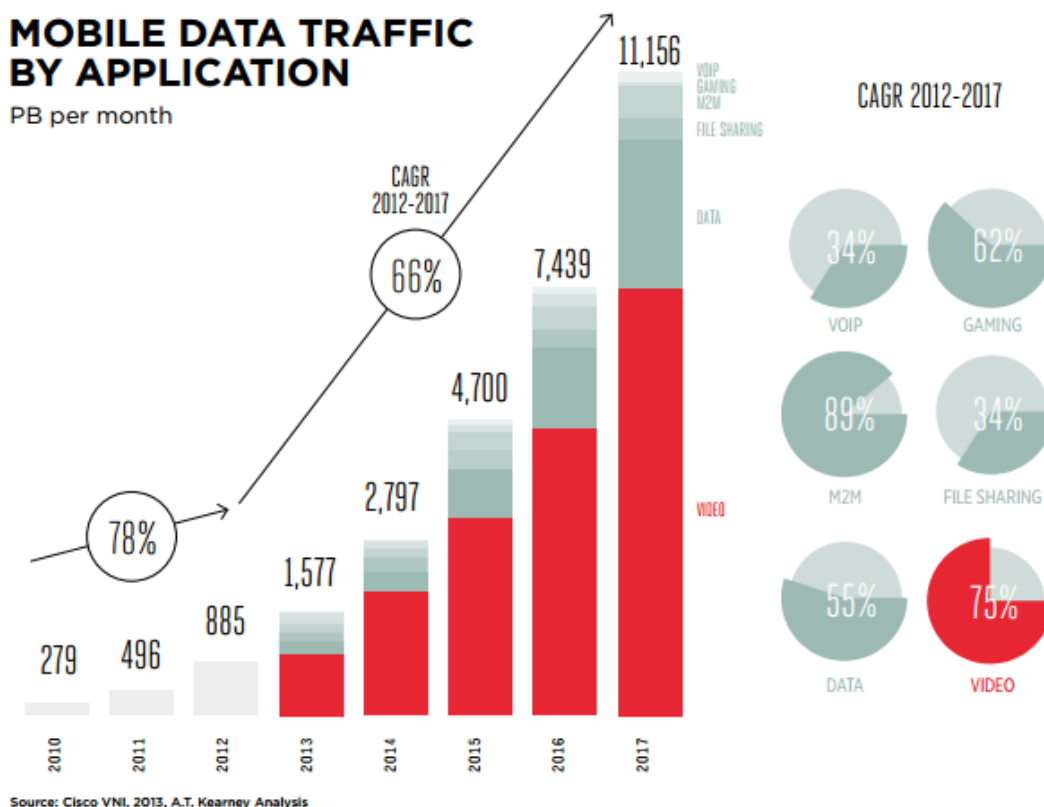


Figure 1.1: 移动应用流量增长趋势

级。尽管更多的用户得到了满足，更多新的商业模式由此孕育而生，这些发展将严重威胁到如今已经接近饱和状态的蜂窝网络。

为了提升蜂窝网络的能力，传统的策略是使用如下方法：

1. 提升发送者到接收者之间的物理层链路的能力。然而，第四代蜂窝网络(LTE-Advanced) 和无线局域网(IEEE 802.11n/ac)所使用的物理层已经将频谱利用率逼近到一个理论的上届，在这方面的更多提升十分的优先。
2. 使用其他的频段。在最好的情况下，我们可以将频段利用率提升到两倍利用率，这远远无法满足对数据需求的增长。
3. 缩小蜂窝的大小进而提升区域频谱利用率。这虽然是一个有效的方法，但是存在的问题是建立新的蜂窝网基站以及提供相应的传输能力的代价太大。

为了提升蜂窝网络的整体吞吐量，研究人员开始注意到设备之间传输(D2D)可能带来的好处。这种传输方式充分挖掘现代移动设备上充足的存

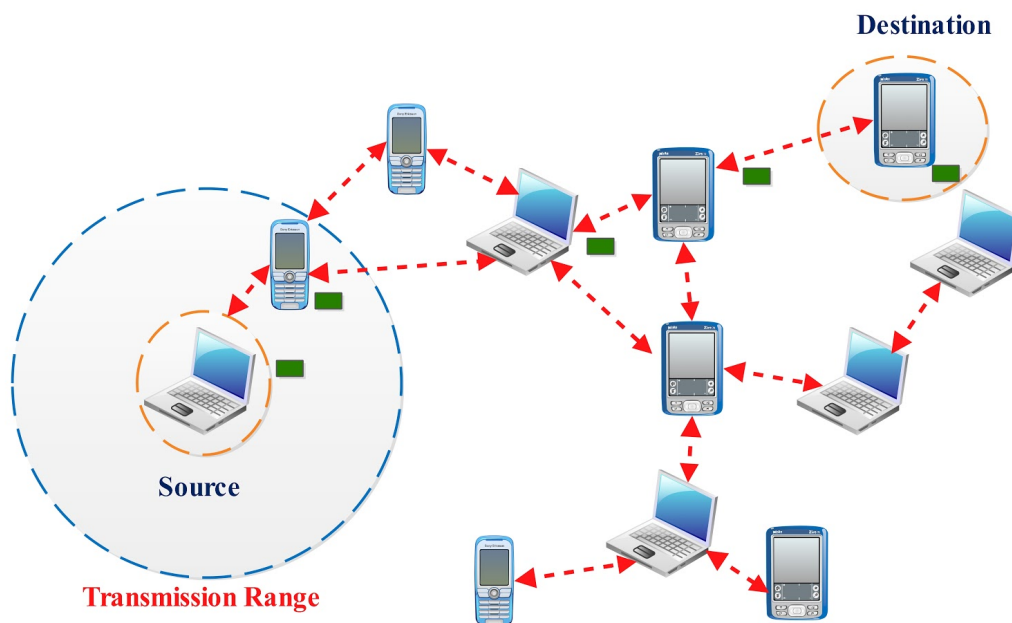


Figure 1.2: Ad Hoc网络示意图

存储空间，每个设备存储一定数量的流行视频，当一个请求发出之后，周围存在缓存的设备就通过D2D链接将视频传输到请求方。因为传输方和接收方之间的距离大多数情况下很短，在一个蜂窝网络中就能够同时存在多个D2D连接，进而带来了频谱效率的巨大提升。

1.2 Ad Hoc 网络

冗余的视频请求和设备之间传输的技术已经成为了提升蜂窝网络吞吐量的主要思路。在D2D扩展而来的Ad Hoc网络是解决传输问题的一种解决方案，Ad Hoc通过在无线网络中建立一个多条网络，让设备之间通信的能力不受限与物理信道，从而扩大无线网络的规模。相比于D2D传输方式，Ad Hoc网络能够承载更大的用户基数，从而提供更多的缓存。

传统的无线传输技术提供设备之间的直接通信(蓝牙和802.11)，直接通信指在无线通信接口的传输范围之内的通信，这就使得一跳的ad hoc网络称为可能。在此之上，人们发觉在无线设备上的多跳传输的可能性。在多跳ad hoc网络中，设备之间通信的目的不仅是为了传输数据，同时还为了转发网络中其他节点的数据。为了达到多跳传输的目的，在无线网络中有一套专门设计的路由协议，来维护各个节点之间的路由信息。

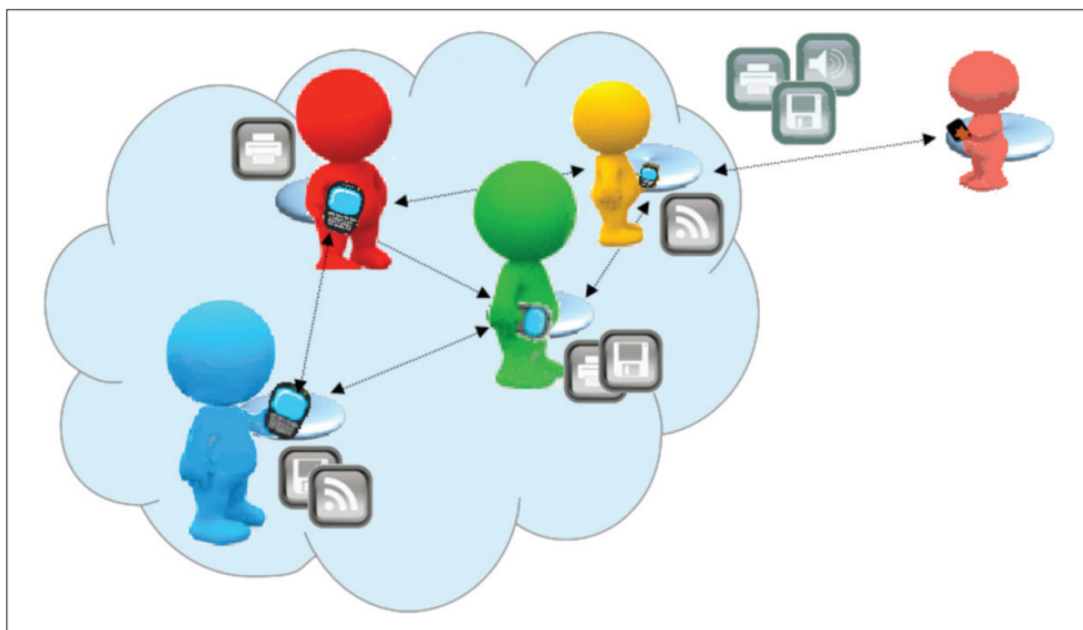


Figure 1.3: 分布式缓存

建立在ad hoc之上, 可以进行各种数据的传输. 典型的传输就包括视频传输, 尽管ad hoc网络所使用的频带与蜂窝网络不同, 但是设备之间的通信必须在相同的频带上进行. 基于此, 在ad hoc网络之中, 各个节点之间相互传输视频资源还是会引发激烈的信道争用. 为此, 传统网络中减少网络拥塞的方法必须再一次应用到ad hoc网络之中. 其中最有效的一个策略就是视频缓存.

1.3 移动视频缓存技术

当移动设备请求互联网上的一个视频之后, 该视频必须从CDN的服务器中取出.CDN帮助减少了互联网的带宽使用, 同时减少了视频播放的延时和抖动. 但是视频必须进一步通过无线运营商的核心网络到达基站, 最终通过无线传输的形式到达用户设备. 无论是运营商的设备网络还是终端所在的无线网络, 都可能成为传输的一个瓶颈. 相关工作已经将在基站上进行视频缓存的策略进行考虑.

然而对于大用户群体的区域,即使提供了基站缓存也无济于事. 根据“Gupta-Kumar”限制[?], 如果有 n 个节点试图进行通信(即形成 $n/2$ 个通信对),每个节点的吞吐量将随着 n 的增加而减少, 衰减的幅度将是 $1/\sqrt{n}$.但是, 如果有足够的内容复用, 使用设备间传输以及缓存的方式, 每个节点将得到不受影响的吞吐量[?].

本文假设的缓存策略即建立在ad hoc网络之上。即，我们假设在视频传输应用之下有一个ad hoc网络作为支撑。因此，任何传输在底层对于我们都是透明的。这样，在网络中用户手中的移动设备就构成了一个分布式的缓存，缓存内容即为用户看过的那些视频。每个用户通过监听网络中的视频数据，能够有效缓存可用的数据。

然而，用户设备的存储空间有限，而视频的数量庞大。因此，决定缓存哪些内容成为了亟待解决的问题。对于互联网视频的缓存，一个有效策略就是使用推荐技术。

1.4 推荐技术

网络商城的出现和发展对消费者的购物习惯有着深远的影响，人们在网络环境下能够访问到不同类别的商品以及每个商品的不同信息。尽管选择的自由性给这个市场带来了巨大的财富，但是这也使得用户选择最适合的产品变得异常困难。解决这类“信息过载”问题主要的方法便是推荐系统。推荐系统通过向用户自动化地给出个性化的推荐，避免用户进行痛苦的筛选。推荐系统已经被用到多个领域，例如推荐书本与CD[? ?],音乐[? ?],电影[? ? ?],新闻[? ? ?],笑话[?],以及网页[? ?]。

推荐问题可以被定义为：根据用户在系统中的历史信息，古迹用户对一个新的物品的反馈，同时向用户提供预测反馈较高的独特有趣的物品的推荐。用户对物品的反馈信息在不同应用领域中各不相同，基本上落入如下三类：标量，二值和单值。标量反馈也叫评分，一般是数值类型或者顺序类型，代表用户对物品所有的认同级别。另一方面，二值反馈只有两种可能的取值，代表对物品的相反的两种态度(喜欢/不喜欢，感兴趣/不感兴趣)。单值反馈记录用户与物品的一次交互信息(例如购买，在线访问)，却没有显式地给出用户对这类物品的认同程度。因为大多数用户倾向于和感兴趣的物品产生交互，单值数据在提供用户偏好这方面还是十分有用的。

在推荐领域的学者面临着几个挑战，这些挑战对它们所提出的算法的性能和使用造成了威胁，这里列出其中最关键的几个：

1. 数据稀疏性。通常情况下，可推荐的物品的总量非常庞大，任意两个用户之间很难存在交集或者仅有几个交集物品。即使整个系统平均的(用户-物品)交互很高，它们通常存在分布不均匀的现象(通常，它们遵循power-law分布[?]或者Weibull分布[?])。因此，用户或者物品的大部分成员只有优先的偏好反馈。因此，一个有效的推荐算法必须将数据的稀疏性考虑在

内[?].

2. 可扩展性. 虽然数据是稀疏的, 但是大多数网站包含着上百万数量的用户和物品. 因此, 一个关键问题就是要考虑计算开销问题, 所设计的推荐算法必须计算开销低或者容易并行化. 另一个解决方案是使用增量式更新的算法, 这类算法伴随着数据的增加, 推荐模型无需整体重新计算, 而是逐渐进行调整[? ?].
3. 冷启动问题. 当一个新用户加入系统, 系统通常缺乏足够的信息为他们提供推荐. 应对这一问题的通常策略是使用混合推荐技术, 结合基于内容推荐与协同数据推荐方法. 另一个方法是在其他网络服务中识别个体用户, 这样针对一个在A网站的冷启动用户, 我们能够根据其在该网站B,C,D中的历史数据进行推荐.

应对这些问题, 一个常用的策略是从多个数据源中构建推荐模型. 在本文中, 我们也试图解决这类问题, 然而不同的是, 我们试图在算法层面上解决问题, 通过充分挖掘数据中可用的信息来达到减少数据稀疏效果.

1.5 本文组织结构

1. 第一章节介绍相关研究背景, 简单介绍了无线视频传输的现状, ad hoc网络, 视频缓存技术以及推荐技术, 并对本文的组织结构进行了介绍.
2. 第二章节介绍视频缓存的相关工作, 重点介绍推荐系统的现有成果.
3. 第三章节介绍我们的视频推荐算法, 我们的算法试图同多个不同反馈现象中得到更精确的推荐模型
4. 第四章节从系统设计层面说明如何将推荐技术应用于流媒体推荐, 同时给出了模拟实验的结果.
5. 第五章节总结工作, 提出下一步研究工作的分析和展望.

第二章 相关工作

2.1 无线视频传输相关工作

当前, 视频请求被当做独立的数据资源请求响应. 每个视频请求会话使用独立的单播传输完成, 不同的视频块之间的传输被视为网页传输, 使用http协议, 并根据网络状况调整视频质量. 这种视频请求方法没有充分利用视频请求的一个特征: 受限的请求模式. 例如, 视频提供商(亚马逊,Netflix)提供有限的视频资源文件, 甚至通过提供免费视频来改变视频请求模式. 简单的广播监听策略在无线视频领域不太有效, 尽管用户请求表现出很明显的可复用性(相同流行视频被反复地请求),对同一个视频请求的异步型太强, 使得两个用户在短时间间隔内请求同一文件的概率几乎为0, 这种特征在视频请求中称为异步内容复用.

多年来, 许多工作都利用这种受限的请求模式提出新的视频传输方案. [??]考虑用户在同一时间请求相同内容(例如视频直播服务),但是不同的用户有不同的信道质量或者视频质量. 这种情况下, 可扩展视频编码能够与某种形式的广播信道编码结合[?].另外一些工作考虑相邻无线用户同一时刻请求相同视频, 相互协作以提高下载链路吞吐量. [?]提出不同用户同时下载相同视频的不同部分, 然后通过D2D传输相互共享内容的方法.

上述工作适用于同步视频请求的场合, 在异步视频请求场合就无法适用. 然而, 如果对每个用户请求独立响应, 将造成系统的瓶颈: 在传统的单播基站传输情况下, 每个用户的吞吐量随着用户数量的增加而线性地下降.

今年的工作表明, 依赖缓存来存储视频资源是有效提高无线传输吞吐量的可靠方法. 从[????]的结果, 我们观察到缓存能够使得吞吐量获得数量级上的提升. 缓存给异步请求提供了很强的复用空间, 同时充分利用了移动设备上尚未完全利用的存储空间. 现有的缓存工作包括如下3类: 1.使用大量专门配置的 " 帮助节点 ", 其中缓存了大量的流行视频, 并且通过短距离传输链路给移动节点提供内容服务. 这些帮助节点一般是含有缓存的基站, 以此来代替优先传输[?] 2.将移动设备上的视频缓存与网络编码数据结合, 来达到视频多播传输的目的[?], 这类方法称为 " 编码多播 " 3.将移动设备缓存与短距离D2D传输结合[?]. 如此以来, 尽管单独的设备只能存储有限的视频资源. 多个设备共同组成一个虚拟缓存, 能够存储大量的视频.

" 编码多播 " 和 "D2D缓存 " 拥有一个相同的有趣性质: 共享虚拟缓存的容

量随着系统中用户数量的增长而线性地增长。这意味着，随着网络中用户数量的增加，聚集的缓存能力也相应增加，这将比缓存内容于基站带来更巨大的收益。

与D2D传输结合设备缓存类似，我们试图做到使设备缓存与ad hoc传输相结合。然而本文的关注重点不在于具体的传输模型，以及节点的移动特性。本文假设任何视频都能够在ad hoc网络中完成视频传输，关注缓存命中率的提升。

2.2 个性化推荐系统

归功于计算机与计算机网络，我们的社会正经受着各方面的转型。我们在网上购物，使用搜索引擎收集信息，并且在互联网上进行着大量的社交活动。这些网络活动和互动都被记录下来，给学者更多研究社会和经济的机，推荐系统也不例外的成为了研究的热点。推荐系统的任务是将用户的数据和偏好转化为对将来行为的预测。推荐系统处于科学与社会经济生活的交叉路口，其巨大的潜力最初由处于信息革命浪尖的互联网企业家发现，并逐渐渗透入各个行业。

当我们对特定用户进行推荐时，基础的方法是选择与该用户相似的其他用户也喜欢的那些物品。即使最简单的方法也能使用多种方式来实现，这是因为推荐领域缺乏能够作出正确推荐的基础原则。例如，我们该如何衡量用户之间的相似性以及如何衡量其不确定性？如何整合来自不同用户的相反的意见？如何处理用户信息缺乏的那些用户？我们是否应该平等地对待每一个数据？这些问题在更复杂的模型里面也同样存在。幸运的是，现实中我们有可以用来测试方法的真实数据，因此，真正评价推荐算法好坏的是实验的结果。

大多数网站提供不同形式的推荐，从简单地将最流行的物品进行推荐到一些复杂的数据挖掘技术。人们很快意识到并不存在唯一的最好的推荐方法。反而，根据当前的应用场景和数据的密度，直接为特定应用而设计的方法更有可能成功。因此，推荐不存在万能药，我们所能做的只是理解推荐前提和推荐机制，然后从实际例子出发解决实际问题。

由于数据存储和处理的代价的下降，推荐系统逐渐扩展到生活中大部分领域。运营商仔细研究着我们的购买记录以推荐其他商品来提高销售量；社交网站分析我们的联系人以帮我们找到新的朋友并使我们在网站上逗留更久；在线广播记下我们跳过的歌曲以更好地为我们服务(更多例子如表??)。一般而言，只要存在大量不同的商品和不同的用户，个性化推荐总能帮助我们将正确的内容推荐给正确的人。推荐对在线网店更为重要，这些公司想利用商品的“长尾效

Table 2.1: popular sites using recommender system

site	what is recommended
Amazon	Books/other products
Facebook	Friends
WeFollow	Friends
MovieLens	Movies
Nanocrowd	Movies
Jinni	Movies
Findory	News
Digg	News
Zite	News
Meehive	News
Netflix	DVDs
CDNOW	CDs/DVDs
eHarmony	Dates
Chemistry	Dates
True.com	Dates
Perfectmatch	Dates
CareerBuilder	Jobs
Monster	Jobs
Pandora	Music
Mufin	Music
StumbleUpon	Web sites

应 "[?] 来提高利益。例如，在亚马逊，20%到40%的产品销售量是来源于不属于前100,000的热销产品[?]。因此，推荐系统对公司的利益有巨大的影响：例如，Netflix中有60%的DVD出租是基于对用户的个性化推荐而来。

2.3 个性化推荐技术

推荐技术在本质上没有优劣之分，随着研究的深入，有更多的推荐技术被发现，使人们面对实际问题时候有了更多的选择，推荐技术大致上可如下分类：

1. 基于内容的推荐技术：被推荐的物品是与用户之前喜欢的物品相似的物

品，这里相似性由物品内容决定，如元数据，关键词等。

2. 协同过滤推荐技术：基于大量用户和物品的历史数据得出的推荐结果，基本分为

(a) 基于记忆的协同过滤：同样是寻找相似的物品或者相似的用户，然后使用相似性作为加权最后得到偏好值

(b) 基于模型的协同过滤：根据所有历史数据训练出模型，随后根据模型作出推荐

3. 混合推荐技术：结合基于内容与基于协同过滤的推荐方式。

本文重点集中于基于模型的协同过滤方法，具体而言，我们关注包含维度下降技巧的推荐技术。维度下降的目标在于降低相关数据的维度，同时保留必要的信息。这类技巧通常被应用于数据挖掘，机器学习和聚类分析等领域。大多数的维度下降方法都包含特征向量抽取，特征向量包含一系列的隐含因子，这些隐含因子描述了历史数据出现模式的一些原因。在观看电影的应用场合，一个观影者可能会考虑电影的如下类别：动作，爱情或喜剧，这些考虑共同构成了该观影者的隐含变量(即隐含特征)，通常由一个多维向量表示。一个包含二维向量(动作和爱情)的特征空间如图??所示，该图显示Peter在动作类电影上有一定的偏好，而Mary更喜欢关于爱情的电影。基于同样的方法，我们可以给电影也赋予隐含变量，之后就能对用户和视频的评分进行定义：两个向量的内积。例如，预期中我们认为Peter不喜欢电影 α ，而更喜欢电影 β ，对于Mary来说正好相反。一旦向量被计算好，推荐就能够很轻松地进行。如果我们使用 K 个隐含因子，那么隐含特征响亮就是 K 维，那么如果 $K(N + M) < NM$ ，那么相关的数据被大量的减少，从而达到维度下降的效果。在实际情况中，此类方法特别适合于存储开销和计算开销巨大的大数据集。

除了使用隐含变量来描述不同的兴趣和类别，用户和物品页可用被赋予单独的类别以达到维度消除的效果。这种情况下，用户-物品的交互数据被用户类别与物品类别之间的关系决定。虽然进行分类的最初目的不是为了维度消除，单类别的数量一般情况下远远小于用户或者物品的数量，从而达到维度下降的效果。

维度下降特别适用于协同过滤场合(这有时也被成为基于模型的协同过滤)，因为一般情况下只有一小部分的用户-物品交互数据可以被观察到，借此方法能大量减少数据量。维度下降有效地减少了推荐的计算复杂度和内存开销，同时

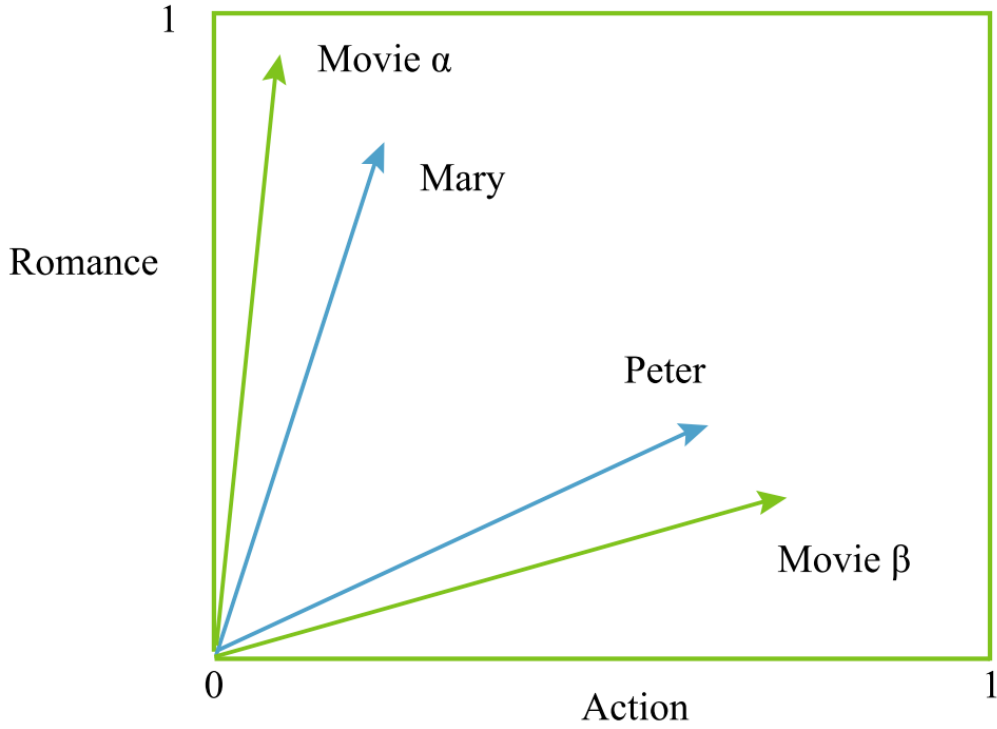


Figure 2.1: 依据隐含变量进行选择的过程

保留了有用的信息。一些有名的维度下降方法包括奇异值分解(SVD)[?],贝叶斯聚类[?],概率型隐语义分析(pLSA)[?],隐含狄利克雷分布(LDA)[?]。

2.3.1 奇异值分解(SVD)

假设有一个 $R \in N \times M$ 矩阵，其中每个元素 $r_{i\alpha}$ 对应着用户 i 对物品 α 的评分(如果对应元素没有评分，那么元素值将为0)。在非评分的系统中，对应的元素值变为 $r_{i\alpha} = 0, 1$ 。那么推荐的过程就是确定当前为0的那些元素中哪个最优可能在将来成为非0元素。对于大部分应用场合， R 都是一个极其稀疏的矩阵。

SVD的降维方法通过使用 K 个隐含变量来对用户喜好和物品特性进行分类，原本的矩阵 R 使用两个矩阵的乘积来估计：

$$R \approx WV \quad (2.1)$$

其中 W 和 V 分别是维度 $N \times K$ 和 $K \times M$ 的矩阵。它们包含着用户的偏好信息以及物品的内容信息，用 K 个隐含向量表示。由于这些隐含变量，SVD属于LSA技术的一种。从公式??中 W 和 V 的乘积来看，物品是根据用户的喜好和物品的内

容在特征空间中的重叠程度来选择的。因为矩阵 \mathbf{R} 被分解称为两个矩阵 \mathbf{W} 和 \mathbf{V} , SVD也被称为矩阵分解方法(MF).

因为矩阵 \mathbf{R} 有些元素是未知的, 我们不能完美的估计原始矩阵. 因此, 目标就变成寻找一个 \tilde{R} 使得其原始矩阵尽量相似:

$$\tilde{R} = \underset{\tilde{R}'}{\operatorname{argmax}} ||R - \tilde{R}'|| \quad (2.2)$$

其中的 \tilde{R}' 被限制为秩为 \mathbf{K} 的矩阵. 由此, SVD提供了衡量两个矩阵 R 和 \tilde{R} 相似性的简单的损失函数. 求解 \tilde{R} , 可以使用基于梯度下降的迭代方法. 矩阵中的每个元素可以使用如下公式估计:

$$\tilde{r}_{i\alpha} = \sum_{k=1}^K w_{ik} v_{k\alpha} \quad (2.3)$$

除此之外, SVD还保有除了额外数据的灵活性. 例如, 我们可以在框架中包含用户对评分带来的偏置项. 假设用户 i 倾向于比其他用户多给平均 b_i 的分数, 而物品 α 比其他物品会多得平均意义上 b_α 的得分, 那么预测分数可以表示为[?]:

$$\tilde{r}_{i\alpha} = b_i + b_\alpha + \sum_{k=1}^K w_{ik} v_{k\alpha} \quad (2.4)$$

除了简单的偏置项, 其他SVD的变体利用用户在社交网络上的关系, 来描述用户喜好之间的相似性, 来使得矩阵分解获得更高的推荐精确度[?].

另外, 我们不仅可以使评分作为唯一的输入数据. 上述方法可以扩展成应用额外的信息[?].假设 $d_{i\alpha}$ 是一个评分对应的额外信息(例如日期).将 $d_{i\alpha}$ 离散化为 l 个正数, 分解成一个 \mathbf{K} 维向量 y_{kl} , 使用矩阵 $\mathbf{Y} \in K \times L$ 描述, 这些额外信息就包含了其与用户喜好和物品内容的关系, 最终可用来预测评分. 例如, 在情人节那天, 很多关于爱情的电影就会大受欢迎, 对应的评分就会变得很高. 有了额外的信息, 评分矩阵就能被表示为:

$$\tilde{r}_{i\alpha} = \sum_{k=1}^K w_{ik} v_{k\alpha} y_{kd_{i\alpha}} \quad (2.5)$$

基于梯度下降的类似推导可以应用到上面, 得到最后的解.

2.3.2 贝叶斯聚类

在贝叶斯网络中，一个变量的取值仅依赖于其父变量。例如，一个变量的概率分布可以用一个条件概率分布 $P(x|pa_x)$ 来描述，其中 pa_x 是变量 x 的父变量。多个独立的变量的联合概率可以被分解为 $P(x_1, \dots, x_N) = \prod_{i=1}^N P(x_i|pa_{x_i})$ ，表示各个变量之间的依赖关系。然而，在贝叶斯网络中最难的部分在于获取最相关的依赖结果。

为了进行个性化推荐，一个简单的实现就是进行两方面的聚类[?]。为了获取一个 $(user, object)$ 对的评分，我们可以将用户和物品分别归入 K_{user} 和 K_{object} 个类别中，与SVD中的 K 一样， K_{user} 和 K_{object} 同样是算法的参数。这里的假设是，在现有数据分布之下，有一个贝叶斯网络控制着其生成过程，其中最简单的假设就是评分 $r_{i\alpha}$ 仅仅依赖于用户的分类 c_i 和物品的分类 c_α 。因此评分 $r_{i\alpha}$ 的概率分布可以被写成：

$$P(r_{i\alpha}) = \sum_{c_i=1}^{K_{user}} \sum_{c_\alpha=1}^{K_{object}} P(r_{i\alpha}|c_i, c_\alpha) P(C_i) P(C_\alpha) \quad (2.6)$$

为了获取 $r_{i\alpha}$ 的估计值，我们需要知道 $P(c_i)$ 、 $P(c_\alpha)$ 和 $P(r_{i\alpha}|c_i, c_\alpha)$ ，因为评分仅依赖于用户类别 x 和物品类别 y ，其等价于求 $P(r|x, y)$ 。

多种方法可以用来求解上述问题，例如使用“置信传播”求取边缘估计或者使用EM算法，抑或使用吉布斯采样来估计概率分布。当联合概率分布很难被采样，而每个变量的条件概率容易采样的时候，吉布斯采样就非常有用。这里我们需要采样的是状态 $(r|(x, y), c_i, c_\alpha)$ ，之后就能用它们来预测评分。具体做法是：

1. 首先对所有用户和物品进行随机的隐含类别分配，得到 N_x 、 N_y 和 N_{xy} ，它们分别对应者处于类别 x 的用户数量，处于类别 y 的物品的数量，以及所有交互数据的类别组合的适量。
2. 然后从 $Beta(N_{xy} + 1, N_x N_y - N_{xy} + 1)$ 中采样一个 P_{xy} ，或者简单地使用 $P_{xy} = N_{xy} / N_x N_y$ 来估计。
3. 同样的，定义 P_x 和 P_y 为一个用户和物品被归类于 x 或者 y 的概率，从狄利克雷分布中采样出 P_x 或者 P_y ，或者简单地使用估计 $P_x = N_x / N$ ， $P_y = N_y / M$ 。

这些 P_{xy} 、 P_x 和 P_y 用来估计系统从一个状态到另一个状态的转移概率。之后，随机挑选一个用户或者物品，对用户使用如下分布更新其隐含类别：

$$P(c_i = x) \propto P_x \prod_{y=1}^{K_{object}} P_{xy}^{\sum_{\alpha} r_{i\alpha} \delta_{c_{\alpha}, y}} (1 - P_{xy})^{\sum_{\alpha} (1 - r_{i\alpha}) \delta_{c_{\alpha}, y}} \quad (2.7)$$

对于物品:

$$P(c_{\alpha} = y) \propto P_y \prod_{x=1}^{K_{user}} P_{xy}^{\sum_i r_{i\alpha} \delta_{c_i, x}} (1 - P_{xy})^{\sum_i (1 - r_{i\alpha}) \delta_{c_i, x}} \quad (2.8)$$

对每个用户和物品更新隐含类别之后, 更新 P_x, P_y 和 P_{xy} 的取值. 经过足够多次迭代, 就可以开始采样用户对物品的评分:

$$\tilde{r}_{i\alpha} = \sum_t P_{xy}(t_c + tT) \delta_{x, c_i(t_c + tT)} \delta_{u, c_{\alpha}(t_c + tT)} \quad (2.9)$$

其中 t_c 和 T 分别代表收敛时间和采样间隔.

这类目标极难优化的方法使用场景十分受限. 例如, 使用吉布斯采样需要很长的收敛时间和一段漫长的采样时间, 实际场景中, 内容的更新速度极快, 模型训练速度必须赶得上更新速度.

2.4 传播学习

随着社交媒体的流行, 用户通常会同时活跃在不同的在线社交网站(OSN)上, 不同的社交网站也对应着不同的服务内容. 例如, 一个用户平时可能使用Facebook与他的好友进行交流, 在Twitter上追踪实时热点事件, 订阅和观看YouTube上的流行视频, 在Yelp上分享和讨论喜欢的餐厅, 等等. 这些跨网络的活动共同组成了一个用户的整体在线足迹, 同时从不同的方面展现了其人口特征和兴趣爱好.

然而, 典型的社交媒体服务通常只在一个在线社交网站上提供. 例如, YouTube的视频推荐服务, 使用的仅仅是用户在YouTube的活动. 使用基于单网络的解决方法的存在着一定的局限性, 用户在一个网络上的活动不足以用来推测出用户的兴趣和实时改变的偏好. 冷启动和稀疏性问题长期以来一直是困扰着实际的个性化服务部署[?]. 因此, 将散落在多个社交网站的用户信息进行收集和学习, 以最终服务于一个目标在线网站, 是解决问题的正确方向.

为了验证跨网络学习的可能性, [?]收集了137317个用户的Google+账号. Google+鼓励用户在账户信息分享其他的网站信息, 作者进一步得到了对应的YouTube, Twitter, Facebook以及Flickr的信息, 发现大部分用户都在多个网站上同时拥有账户, 如表??和表??.为了验证在不同平台上, 用户拥有不同的活跃

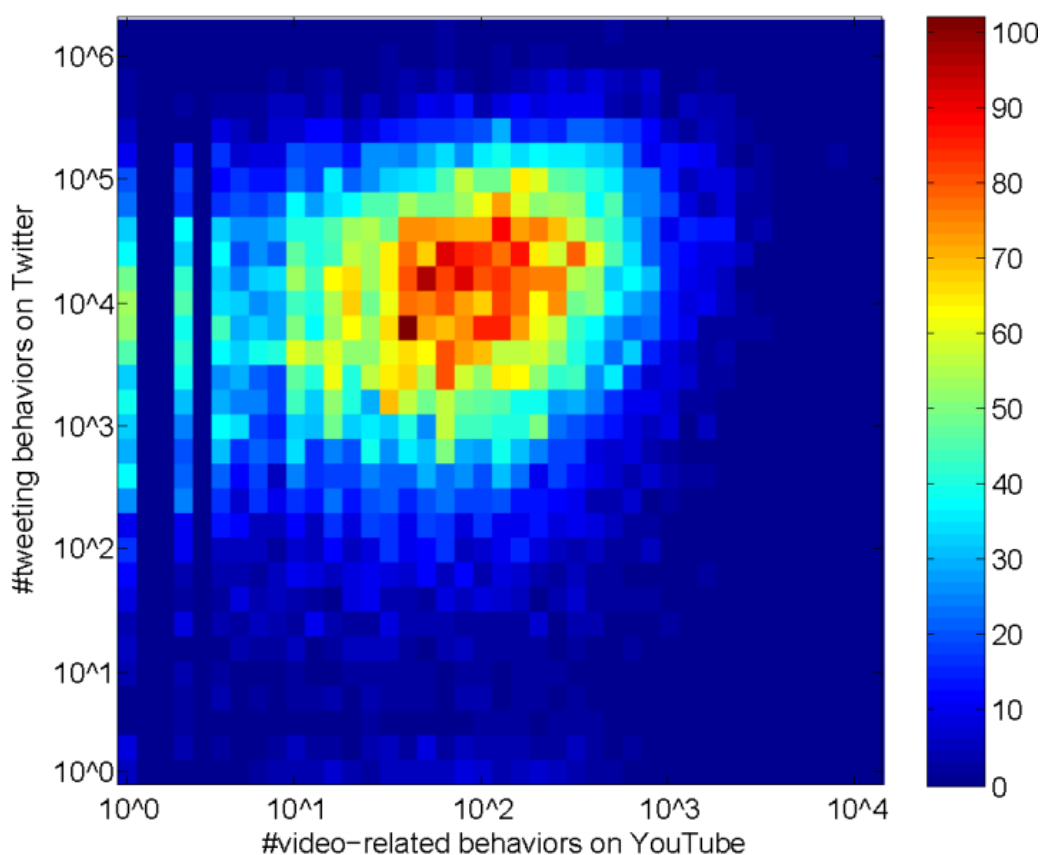


Figure 2.2: 用户在YouTube和Twitter上活动计数的热度图

度，作者比较了所有用户在YouTube和Twitter上的活动数量。如图??，每个用户都在图上拥有一个坐标，该图显示了所有用户活动数量分布的热度图，色相越接近红色代表该区域内的用户数量越多。图中显示：1. 红色的区域大量集中于对角线的左上角区域，这说明用户在Twitter上的活跃度远高于用户在YouTube上视频活动的活跃度。2. 对于只有少量视频相关活动的用户，如在x轴上 $10^0 - 10^1$ 的区域，可见的twitter活动情况在范围 $10^2 - 10^5$ 内变化。这说明在YouTube上的冷启动和数据稀疏问题可以借助Twitter上的辅助用户数据来解决。

Table 2.2: 用户在其他网站上的账号统计

	YouTube	Twitter	Facebook	Flickr
#account	52,390	43,772	32,020	12,242
proportion	0.3815	0.3188	0.2259	0.0892

Table 2.3: 用户在四个网站的重叠度

	YouTube	Twitter	Facebook	Flickr
YouTube	1	0.4253	0.3109	0.1294
Twitter	0.5090	1	0.5376	0.2223
Facebook	0.5251	0.7586	1	0.2207
Flickr	0.5537	0.7948	0.5591	1

一句话概括传播学习(transfer learning), 那就是使用跨网络用户建模方法, 关注多个网络数据的整合. 在[?]中, 作者使用用户在Flickr, Twitter和Delicious中的个人信息来解决推荐系统中的冷启动问题. Deng等人提出个性化YouTube视频推荐解决方法, 其中就利用了用户在Google+中的个人信息[?]. 在[?]中, 作者使用twitter的实时性和社交性来辅助YouTube中的视频应用. 在[?]中, 维基页面中的权威信息被用来辅助Twitter上的事件检测.

这类跨网络协同应用的一个关键问题是如何获取多个网络中对应的账号信息, 排除特殊情况, 一般情况下这类特殊关系很难得到公布和使用, 因此这种方法成为了少数内部机构可以享用的特权方法. 本文在试图使用这类方法时候也遇到此类问题, 因此使用了不同的信息渠道: 从一个网络中得到不同类型的信息, 使用传播学习的方法整合信息, 已达到推荐的效果. 虽然使用的数据源不同, 却面临着相似的问题:

1. 不同网络中的数据存在的异构性, 而同一个网络中不同信号的数据也是异构的, 如何将异构数据服务于相同的目标是一个问题.
2. 异构的数据相互之间可能存在冲突, 向目标的相反方向靠近, 做好不同数据之间使用的平衡和对齐是一个问题.

总而言之, 关键点在于保留数据的目标共性而排除数据间的本质差异性.

第三章 基于多信号融合的分解推荐模型

3.1 推荐系统简介

推荐系统是视频分享网站的一个重要组成部分。最初，它的动机是为了提高用户体验。随着视频观看数量在互联网上的增长，推荐正在被其他领域使用，例如，内容传输，内容缓存等类似的流媒体领域。对于普通用户而言，推荐使他们从“信息过载”的痛苦中拜托出来；对于网站运营商，推荐确保它们的用户能够在网站上待得更久，从而为网站带来更多的利润。

推荐系统的核心目标是揣测用户的选择或者偏好。在众多的推荐技术中，基于模型的协同过滤方法脱颖而出。它的基本想法是：用户和视频的交互行为受到一组隐藏的因子所控制，而这些隐藏的因子可以从用户—视频现有的交互数据中推测而来。

大多数的推荐任务都面临着“数据稀疏”和“冷启动”的问题。当用户或者视频在网站上留下的记录有限，为他们训练出来的模型将会变得不精确，给出的推荐结果也变得十分糟糕。对于视频网站而言，老用户的离开和新用户的加入发生的十分频繁；并且每天都有大量的新鲜视频会更新。因此，视频网站中总会存在大量的“冷”用户和视频。尽管有许多模型试图对用户行为作出更好的解释和预测，当面对“冷启动”问题时，大多数方法都显得力不从心。

当前，人们普遍的共识是“更多的数据优于更复杂的模型”。相关工作将注意力转移到利用更多的数据来提高推荐效果。这类方法中通常存在多种形式的反馈，因此针对不同形式的反馈会有不同的建模方式。通过某种共享机制，这些数据共同被用来训练一个模型。

[?]使用LDA从用户的评论文本中提取商品的相关主题分布，同时从购买记录中提取用户和商品在购买行为上的隐含特征，通过将主题和隐含特征相关联，得到了可解释的推荐结果。[?]通过关联用户在视频网站和tweeter上的数据，从tweeter文中提取用户的兴趣分布，从而更好地指导该用户在视频网站上的推荐。

为了达到现有推荐技术的效果，关键在于建模出一组精确的隐含因子，通常将其形式化地表示为一个 d 维度的特征向量，这些因子从用户的反馈数据中习得。为了构建隐含因子模型，使用反馈数据的方式是其中的关键部分。为了建模隐式用户反馈，[?]提出了基于偏序对的方法“BPR”，它将用户的反馈视为正

例反馈，而将未交互的视频作为负例反馈。通过最大化所有“正例反馈的偏好大于负例反馈”的概率，[?]为每个用户提供了一个为视频排序的模型。我们称这类反馈数据为“偏序信号”。“BPR”模型试图为每个用户学习一个视频在偏好值上的偏序关系。

推荐的目的是推测用户对物品的偏好，偏序关系是偏好的重要的一面。然而，很多时候单纯使用偏序信号不足以挖掘出用户视频的特征，尤其是对于偏序反馈缺乏的冷视频而言更是如此。因此，我们考虑使用与偏序信号相关的其他反馈数据。在很多场合下，数据通常表现出一定程度的聚集效应，例如在视频网站中用户的连续观看的视频所呈现的聚集性。如果这类反馈与偏好相关且能够反映用户的行为，那么它们将给推荐结果带来提高。

在视频推荐相关的文献中，基于最近邻的启发算法一直是现实应用场合的一个简单且有效的方法。典型的系统部署是YouTube推荐系统[?]。它通过对所有视频建立一个“共视图”，在图上应用基于图的推荐算法得到视频之间的相关性。根据用户当前观看的视频，给出与该视频相关度最高的那些视频。其中的基本想法是：在同一次会话中，用户更倾向于观看同一类型的视频，那么这些视频在某种程度上而言，是相似的。

我们将视频的“共视”当做一种聚类的信号。这类信号被YouTube用户提供相关视频推荐，但是YouTube对每个用户做了一个全局性的推荐。为了将聚类信号结合到一个个性化的框架中，我们同样要进行个性化聚类的建模。这里的难点在于：

1. 如何对聚类信号进行个性化的建模
2. 如何将不同的建模方式整合到一个推荐框架中

从[?]得到启发，我们使用隐含因子模型对偏序信号和聚类信号进行建模，将推荐问题形式化为单类分类问题。因此，框架中存在两种信号，它们共同训练一个目标，不同信号的模型相互耦合且共享特征向量。

3.2 相关工作

互联网承载着数以亿计的视频内容，在如此多的选择之下，用户极易变得无所适从。对于视频网站而言，良好的推荐系统是它们赖以成功的关键因素。在YouTube中，提供了相关视频的推荐服务。[?]发现在所有的YouTube的用户流量中，大部分都来自“相关视频”的推荐服务。这说明，大部分人已经十分地依赖于推荐系统的功能了。

为了做出推荐，YouTube需要找出与当前观看视频最相似的视频。[?]揭示了YouTube所使用的推荐机制，其本质上是使用一个简单有效的 " 共视计数 " 作为相似性的指标，对于每一对在同一次会话中被一起观看的视频，都算一次计数。[?]扩展了 " 共视 " 的概念，在此之上应用随机游走算法得出更精确的相似度计算。

另一方面，学者在个性化推荐领域也提出许多方法解决推荐问题。Netflix大赛证实了基于隐含因子的分解模型在效果上要优于经典的近邻算法[?]。为了进一步解决隐式反馈数据问题，[?]进一步提出了基于偏序关系的优化方法。在此之上，[?]发现用户的偏好呈现局部性，提出了基于局部偏好排序的系统过滤。[?]扩展了传统的二维的矩阵分解，使用三维的张量分解在上下文感知的场景下做推荐，这类方法的一个明显缺点是：增加的维度会明显增加模型的复杂度，从而产生 " 过拟合 " 问题。因为构建复杂模型容易造成 " 过拟合 " 问题，更多的学者倾向于专注减少数据的稀疏性。

为了解决数据稀疏性问题，" 共分解模型 " 是普遍采用的一种手段。其基本架设同样是隐含因子，只不过不同的数据网络中的反馈由一组相关的隐含因子决定，这样达到的效果是从多个网络中学习共同的一个目标。[?]假设用户在YouTube上的活动与其在Tweeter上的活动相关，基本动机是用户在Tweeter上的活动相较于在YouTube上要活跃很多，从Tweeter上得到的与用户相关的信息有助于其在YouTube上的推荐。[?]试图将购物网站上的隐含特征与商品的评价文本上的主题分布相互对齐。[?]将用户的GPS信息融合进基于位置服务的推荐系统。如果原始数据网络只能提供优先的反馈数据，一些相关的数据网络能够用来弥补数据缺失带来的不足。

在本章的讨论中，我们要将我们的模型建立在 " 共分解 " 技术之上。跟之前工作不同之处在于，我们的方法将使用同一个数据网络中的数据的不同方面。这样做的优点在于很自然的保证了隐含因子的一致性（用户在不同网络中的兴趣分布往往会不同）。我们的模型同时使用 " 偏序信号 " 和 " 聚类信号 "，对于聚类信号，我们将其融入个性化推荐的框架。

本章的贡献如下：

1. 我们通过在同一个数据源中抽取两类不同的数据信号，以解决推荐系统中普遍存在的数据稀疏性问题。
2. 我们将两种不同类型的信号融合到一个同一的推荐框架之中。
3. 我们使用真实场景下的数据进行了实验，并且验证了方法的有效性。

3.3 排序问题

3.3.1 形式化定义

给定一个用户和一系列视频，其中一些视频已经被该用户观看过，另外一些没看过。推荐的任务是从未看过的视频中选择一些视频出来，使得这些视频在将来尽可能多的被该用户观看。推荐模型针对每个（用户，视频）对，会给出该用户对视频的偏好值的一个评分。这样，每个用户对所有视频有一系列的评分，根据这些偏好值评分给推荐提供指导。

3.3.2 隐含因子模型

每个用户对每个视频都会有一个偏好值评分，该评分由一个评分函数 $h(j|i)$ 给出， i 代表用户， j 代表视频，该评分函数就由模型假设的来。其中一种建模方法是隐含因子模型，其中因子表示与评分相关的所有因素，通常由一个 D 维的向量表示。例如，在视频推荐领域，假设维度 $D=2$ ，那么就代表此时用户对视频的评分就由两个因素来决定。对于视频来说，它们可能描述了该视频的喜剧成分和动作武打的成分。对于用户而言，它们就对应着用户对喜剧的喜爱程度以及对动作武打片的喜爱程度。相对应的评分函数如下：

$$h(j|i) = w_i + w_j + u_i^T v_j \quad (3.1)$$

其中 w 代表偏置项，而 u ， v 则分别对应着用户和视频的隐含特征。

ii 隐含特征的图示

3.3.3 基于偏序对的学习

基于上述的模型假设，模型参数可以从用户的反馈数据中得来。视频网站中的反馈数据一般以隐式反馈形式出现，处理隐式数据的有效方法是采用基于偏序对的学习方法。基本想法是：对于每个用户 i ，有观看过的视频 $j \in V^+(i)$ 和没看过的视频 $k \in V^-(i)$ ，那么 " i 对于 j 的偏好值要高过对 k 的偏好值"：

$$j \succ_i k \Leftrightarrow j \in V^+(i) \wedge k \in V^-(i). \quad (3.2)$$

通过这些处理，我们得到一批偏序对数据 $S_D \succ U \times V \times V$ ：

$$\langle i, j, k \rangle \in S_D \Leftrightarrow j \in V^+(i) \wedge k \in V^-(i). \quad (3.3)$$

有了这些偏序对反馈数据之后，数据与模型的关系通过定义偏序对产生的概率从而建立起来：

$$p(j \succ k) := \sigma(\hat{h}(j|i) - \hat{h}(k|i)) \quad (3.4)$$

其中, h 是评分函数, 最终的目标就是使得数据集的联合概率最大:

$$\operatorname{argmax}_{\Theta} \prod_{(i,j,k) \in S_D} p(j \succ_i k), \quad (3.5)$$

因为 S_D 中包含着大量的数据, 一般采用随机梯度下降的方式来训练参数. 在每一轮的更新中, 对于每个用户观看过的视频 j , 都从没看过的视频中采样一个视频 k , 然后使用梯度下降进行更新:

$$\theta \leftarrow \theta - \eta(1 - \sigma(\hat{h}(i, j) - \hat{h}(i, k))) \frac{\partial}{\partial \theta} (\hat{h}(i, j) - \hat{h}(i, k)). \quad (3.6)$$

其中 η 是学习速率. 这个框架被提出以专门处理隐式反馈数据, 命名为 " 贝叶斯个性化排序 " .

3.4 聚类的引入

给定一系列视频 V 和有限的聚类 C , 标准的聚类任务将会把每个视频分配到其中一个聚类 $c \in C$ 中. 然而, 我们的目的不在于做显式的聚类划分, 而仅仅是利用聚类现象中的信息. 具体而言, 我们打算利用同 YouTube 中一样的聚类信号, 将用户在视频在观看行为上的聚类现象作为聚类的一种反馈. YouTube 将聚类计数作为视频相似度的一种衡量, 越高的计数值代表越高的相似度, 这种做法的结果是将所有的视频做了个全局性的聚类. 然而, 我们的目的是将这种聚类反馈结合进个性化推荐的框架中, 也就是说, 在利用信号方面, 我们希望能达到个性化聚类的结果, 即, 对不同的用户, 会有不同的聚类产生.

3.4.1 聚类模型

聚类是一个十分主观的任务, 不同的人会表现出不同的聚类标准. 例如, 有些人会认为 '哈利波特' 系列是属于小孩子的电影, 但是另外一些人会认为它仅仅是一类魔幻系列的电影, 同样适合成年人观看. 为了完成一次聚类任务, 我们需要知道用户在聚类上都采用哪些标准以及它们的倾向度如何, 而视频在这些标准上都体现出怎样的特性. 但是直观上而言, 不论用户和视频在这些标准上展现出怎样的差异性, 它们都是由一套标准所决定, 不同的地方仅仅在于一个 " 度 " .

为了将一系列视频进行聚类, 常见的思维模式是 " 相较于其他视频, 某个视频和另外一个视频更适合聚类在一起 " . 这也就是说, 我们的第一步就是将视频进行两两聚类. 然而, 我们的目标仅仅在于拟合聚类信号而非真实地进行



Figure 3.1: pairwise cluster extraction

聚类，能够进行两两聚类已经足够。到目前为止，我们的目标变成了为聚类寻找一个合适的评分函数 $f(j, k|i)$ ，其中 i 表示用户而 j, k 代表视频。

类比隐式因子协同过滤方法，我们假设在聚类信号之下存在一组控制聚类行为的隐含因子。具体而言，如果 D 代表隐含因子的数量，那么每个视频由一个隐含向量 $\theta \in R^D$ ；为了关联两个视频，我们将每个用户由一个关联矩阵 $\Lambda \in R^{D \times D}$ 表示。这个矩阵用来描述用户认为不同视频之间的两个因子在聚类上有多大的相关性。由此，聚类的评分函数定义如下：

$$\hat{f}(\langle j, k \rangle|i) = \theta_j^T \Lambda_i \theta_k \quad (3.7)$$

我们做了进一步的简化，将关联矩阵假设为对角矩阵，其中的假设是一个隐含因子只会与对应的另一个隐含因子相关。为此，关联矩阵简化为了 D 维的关联向量。这样，我们的聚类模型就与排序模型达到了一致。

3.4.2 聚类信号的截取

如之前所言，我们将视“共视”行为作为聚类反馈。在视频网站里，用户的观看行为都会有对应的时间戳对应。对每个用户，我们将视频在时间轴上排序，在时间轴上维护一个时间区间为 δt 的移动时间窗口。对于在时间窗内的任意两个视频 $\langle j, k \rangle$ ，我们取出三元组 $\langle i, j, k \rangle$ 作为一个聚类信号，如??所示。

δt 的取值决定了对聚类的阈值以及所能获取的聚类信号的数量。我们不能将 δ 设置得太大，如24h。因为太大的时间窗会将多次会话中的观看行为归入一次聚类中，造成聚类信号中混入大量噪音；我们同样不能将其值设置的太小，因为那样将得不到足够的聚类信号。

为了在精确的时间窗口和足够的反馈信号之间做权衡，在保证时间窗口足够小的同时，我们还要保证聚类信号足够多。因此，我们从小窗口开始，寻找一个反馈数量增加最慢的截断长度。经过多次尝试，我们选择设置 $\delta t = 2\text{hour}$ 。因为两次点击的间隔一般都会在1个小时之内，一系列的点击一般都集中在2小时之内。

3.4.3 聚类信号的优化

为了与排序信号的优化保持一致，我们将聚类建模为一个分类问题，一般情况下是需要正例和负例作为反馈的。然而，在反馈并非显式聚类反馈时候，我们容易定义正例聚类，却很难定义出负例聚类。例如，对于我们要研究的视频网站，用户通常会在一次会话中观看一系列视频，根据用户在一次会话中观看相似的视频的直觉，我们可以很容易地将这些视频视为一个正例聚类。但是，如果两个视频不再同一次会话中被同时观看，我们就不能说这两个视频对于该用户而言属于不同的聚类（负例聚类），因为太多其他因素会造成这种观看时间上的偏移，例如：观看时间的限制；视频没有被发现；视频发布的时间不同。

受[?]的启发，我们为每个正例聚类 $\langle i, j, k \rangle$ 随机采样未观看过的视频 l ，为事件" $\langle j, k \rangle$ 相较于视频对 $\langle j, l \rangle$ ，用户 i 更容易将 $\langle j, k \rangle$ 视频对聚类在一起"建模，事件的概率为：

$$p(\langle j, k \rangle \succ \langle j, l \rangle) := \sigma(\hat{f}(\langle j, k \rangle | i) - \hat{f}(\langle j, l \rangle | i)), \quad (3.8)$$

其中 f 代表聚类的评分函数。所有这些四元组 $\langle i, j, k, l \rangle$ 组成了我们的聚类信号的数据集 S_Y 。聚类的最终目标是最大化数据集的联合概率：

$$\operatorname{argmax}_{\theta, \Lambda} \prod_{(i, j, k, l) \in S_Y} p(\langle j, k \rangle \succ \langle j, l \rangle), \quad (3.9)$$

3.5 整合模型

3.5.1 动机与直觉

在模型介绍之前，我们将就为何将偏序信号和聚类信号结合起来作出一个直观的解释。其中的关键思想是这些信号共享一组相同的参数，聚类信号和偏序信号共同用来训练一个模型，当偏序信号缺乏时候，这给聚类信号帮助模型训练提供了一个机会。

如图??所示，视频 x 与视频 s, t, y, z 紧紧地关联在一起(每次 x 的出现都伴随着视频 s, t, y, z 的出现)，但是相对于所有的数据， x 的训练数据相对较少。假设我们要对某个观看过视频 s, t, y, z 的用户做推荐，因为视频 x 的高度相关性，我们希望 x 会被推荐。如果仅提供偏序信号，那么 x 很难被推荐到，因为其他流行的但却相关性不强的视频会将 x 推到底部。为了提升视频 x 的位置，我们需要告诉系统视频 x 很重要。聚类信号通过在视频 x 和流行视频 s, t, y, z 之间建立关联使得 x 趋近于这些流行视频，从而间接地提升了视频的权重，进而增加了向该用户推荐该视频的概率。

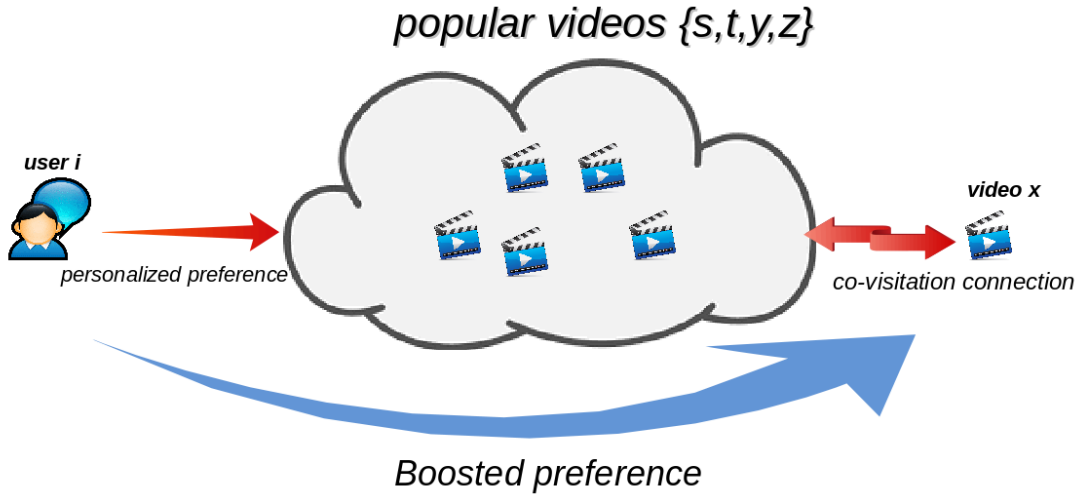


Figure 3.2: clustering boosted preference

偏序信号体现出个性化的流行度的趋势，这意味着它会更偏向于流行视频而不是相关视频，相关视频不一定是流行的，而是主题兴趣十分相近的内容。而聚类型号更加关注相关度这一因素，这也解释了为何聚类信号在YouTube上的成功应用。与YouTube不同的一点是，对于不同的用户，一个视频的相关视频总会是一样的；而我们描述的相关视频对不同的用户将会有不同的视频列表。

3.5.2 信号的整合

图??解释了框架的整合过程。对于偏序信号，我们区分观看过的和没观看过的视频，截取出三元素 $\langle user, observed, unobserved \rangle$ 。为了将其建模为分类问题，我们使用了偏序模型和逻辑回归得到三元组这一事件的概率。因此目标函数就是所有这些事件的联合概率，其中的参数涉及用户和视频的偏序隐含因子向量；对于聚类信号，我们使用时间窗口来过滤出视频聚类，针对两两视频聚类进行建模。因为我们假设用户的关联矩阵是对角矩阵，那么用户和视频同样可以用一个聚类隐含因子向量表示。通过逻辑回归和聚类模型我们得到聚类反馈事件的概率。进一步假设聚类是相互独立的，同样得到数据集的联合概率。同样的，我们假设偏序事件和聚类事件之间相互独立，如此就可以将两个目标整合进一个目标函数。

到目前为之，我们有偏序反馈和聚类反馈，通过隐含因子模型将其转换成两个目标，偏序反馈和聚类反馈各有一套隐含向量。最初我们尝试将这两组隐含向量关联，通过使用模最小化来使得两组向量尽量相似，但是这样做的结果

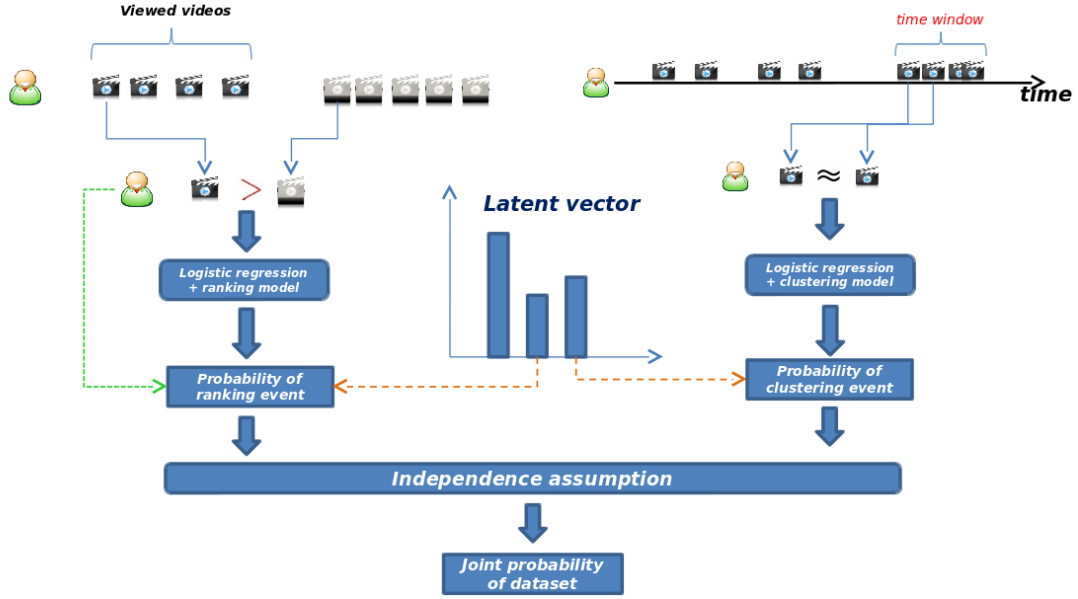


Figure 3.3: integration process

并不理想。最初引入聚类信号的动机是弥补数据的稀疏从而减缓“过拟合”问题。然而，另外一组聚类参数使得模型又将参数的个数增加了，因为聚类反馈数量与偏序反馈数量相当，这使得“过拟合”问题更加严重。为了降低模型的复杂度，我们决定让两组参数更加耦合且让两组反馈信号使用一组隐含特征向量，这为效果带来了很大的提升。

3.5.3 模型描述

图??是模型的图形化表示，其中的关键部分是：为了产生两类随机事件，只与一组隐含因子向量随机变量相关。换句话说，一旦隐含用户和视频的隐含因子向量随机变量被确定，两类事件的分布就已经确定。图??描述了各个随机变量以及超变量之间的依赖关系，符号在表??列出。

框架的生成过程如下：

1. 对每个用户*i*

- (a) 从分布 $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$ 中采样一个用户隐含向量 u_i
- (b) 从分布 $v_j, v_k \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$ 中采样视频的隐含向量 v_j, v_k

2. 对于每个三元组 $\langle i, j, k \rangle \in S_D$ ，采样一个随机变量 $\delta_{ijk} = 1 \sim \text{Bernoulli}(\rho_{ijk})$

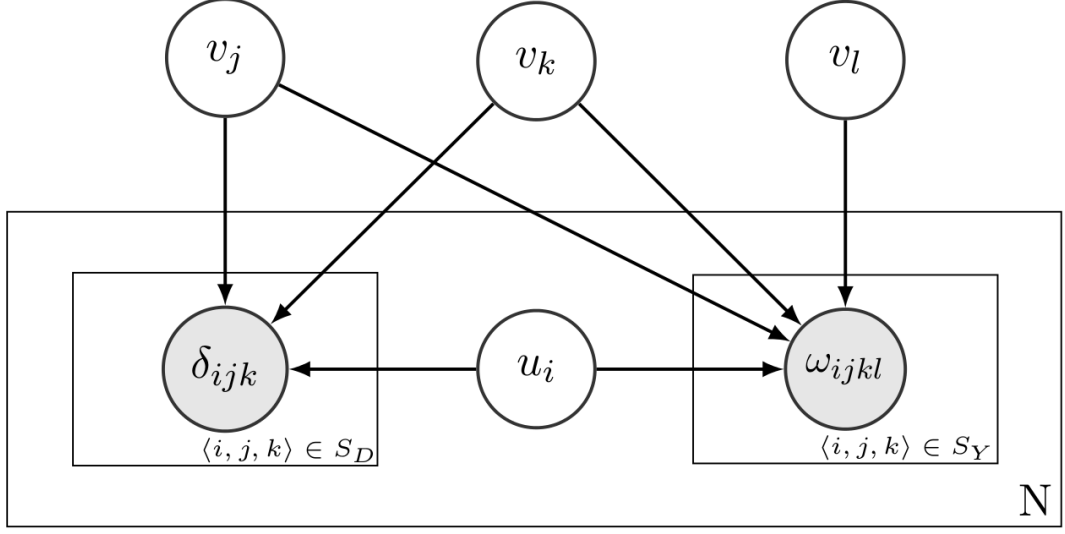


Figure 3.4: graphical model of ClusterRank

其中 ρ_{ijk} 是 " 用户 i 对 j 的偏好大于对 k 的偏好 " 的概率:

$$\begin{aligned}
 \rho_{ijk} &= P(\langle i, j \rangle \succ \langle i, k \rangle | u_i, v_j, v_k) \\
 &= P(\delta_{ijk} = 1 | u_i, v_j, v_k) \\
 &= (1 + \exp^{-(\hat{h}(u_i, v_j) - \hat{h}(u_i, v_k))})^{-1}
 \end{aligned} \tag{3.10}$$

3. 对于每个聚类信号 $\langle i, j, k, l \rangle \in S_Y$, 采样随机变量 $\omega_{ijkl} = 1 \sim \text{Bernoulli}(\varphi_{ijkl})$ 其中 φ_{ijkl} 是 " 用户 i 倾向将 $\{v_j, v_k\}$ 聚类的程度高于将 $\{v_j, v_l\}$ 聚类 " 的概率:

$$\begin{aligned}
 \varphi_{ijkl} &= P(\langle i, j, k \rangle \succ \langle i, j, l \rangle | u_i, v_j, v_k, v_l) \\
 &= P(\omega_{ijkl} = 1 | u_i, v_j, v_k, v_l) \\
 &= (1 + \exp^{-(\hat{f}(u_i, v_j, v_k) - \hat{f}(u_i, v_j, v_l))})^{-1}
 \end{aligned} \tag{3.11}$$

其中 $\hat{f}(u_i, v_j, v_k, v_l) = \sum_{d=1}^D u_{i,d} v_{j,d} v_{k,d}$

通过贝叶斯推导, 我们得到如下后验概率:

$$\begin{aligned}
 P(\mathbf{U}, \mathbf{V} | \boldsymbol{\delta}, \boldsymbol{\omega}, \lambda_u, \lambda_v) &\propto \\
 P(\mathbf{U} | \lambda_u) P(\mathbf{V} | \lambda_v) P(\boldsymbol{\delta} | \mathbf{U}, \mathbf{V}) P(\boldsymbol{\omega} | \mathbf{U}, \mathbf{V})
 \end{aligned} \tag{3.12}$$

Table 3.1: notations

U	数据集中的所有用户集合
V	数据集中的所有视频集合
N	用户的数量
M	视频的数量
$u_{i,d}$	用户隐含向量的第d个因子
$v_{j,d}$	视频隐含向量的第d个因子
w_i	用户(或视频)偏置项的第i个因子
δ_{ijk}	用户i喜欢j多于k这一事件
ω_{ijk}	用户i偏向 $\langle j, k \rangle$ 多于 $\langle j, l \rangle$ 组成一个聚类这个事件
S_D	所有的偏序数据集
S_Y	所有的聚类数据集
α_u	偏序数据的用户正则化项
α_v	偏序数据的视频正则化项
β_u	聚类数据的用户正则化项
β_v	聚类数据的视频正则化项

使用最大化后验概率方法，我们得到如下目标函数：

$$\begin{aligned}
 \mathcal{L} = & - \underbrace{\sum_{\langle i,j,k \rangle \in S_D} \log(1 + \exp^{\Delta \hat{h}})}_{\text{ranking data likelihood}} \\
 & - \underbrace{\sum_{\langle i,j,k,l \rangle \in S_Y} \log(1 + \exp^{\Delta \hat{f}})}_{\text{clustering data likelihood}} \\
 & - \frac{\lambda_u}{2} \sum_{i=1}^n u_i^T u_i - \frac{\lambda_v}{2} \sum_{i=1}^m v_i^T v_i
 \end{aligned} \tag{3.13}$$

其中 $\Delta \hat{h} = \hat{h}(i, j) - \hat{h}(i, k)$ 且 $\Delta \hat{f} = \hat{f}(i, j, k) - \hat{f}(i, j, l)$.

3.5.4 参数学习

公式 (??) 包含两个组成部分，一个与偏序信号相关另一个与聚类信号相关。里面包含两个要优化的目标，通过贝叶斯推导，它们被整合入一个目标中。这类目标函数一般使用随机梯度下降方法(SGD)优化。通常对于包含多种优化目标的函数，一般做法是一步步分别优化：先优化其中一个目标，然后使用优化

后的参数优化另一个目标，如此迭代进行

若严格遵从概率模型下的建模，那么两个信号被惩罚的程度一样。为了区别不同信号对模型的贡献，我们选择对不同的信号采用不同的惩罚参数，这样做可以在参数调节上有更高的自由度。实际情况中，对聚类信号实施更轻的惩罚能够获得较高的效果提升。

区别对待两类信号可以有如下解释：

1. 聚类信号的数量相较于偏序信号数量少许多，且聚类信号的梯度比偏序信号小许多。在我们的实验中，当算法收敛之后，偏序信号和聚类信号的梯度大小分别为0.013和0.005。
2. 一个聚类信号比一个偏序信号包含更多有用信息。正如之前所述，"冷视频"问题是难以避免的。对于这些视频，有限的偏序信号让模型过度扭曲向这些包含记录的个体。与该问题相反的是，这些"冷视频"一般会与流行视频在一次会话中被观看。这就是说，冷视频总会与流行的视频相关联，使用这种聚类关联让这些冷视频的特征跳跃到一个精确的状态(因为这些流行的视频已经十分精确)

这样，新的目标函数如下：

$$\begin{aligned}
 \mathcal{L}^{(2)} = & - \underbrace{\sum_{\langle i,j,k \rangle \in S_D} \log(1 + \exp^{\Delta \hat{h}})}_{\text{ranking data likelihood}} \\
 & - \underbrace{\sum_{\langle i,j,k,l \rangle \in S_Y} \log(1 + \exp^{\Delta \hat{f}})}_{\text{clustering data likelihood}} \\
 & + \mathbb{1}_Y(\langle i,j,k \rangle) \left(-\frac{\alpha_u}{2} \sum_{i=1}^n u_i^\top u_i - \frac{\alpha_v}{2} \sum_{i=1}^m v_i^\top v_i \right) \\
 & + \mathbb{1}_D(\langle i,j,k,l \rangle) \left(-\frac{\beta_u}{2} \sum_{i=1}^n u_i^\top u_i - \frac{\beta_v}{2} \sum_{i=1}^m v_i^\top v_i \right)
 \end{aligned} \tag{3.14}$$

其中 $\mathbb{1}_D(x)$ 和 $\mathbb{1}_Y(x)$ 的定义如下：

$$\mathbb{1}_D(x) = \begin{cases} 1 & : x \in S_D \\ 0 & : x \notin S_D \end{cases} \quad \mathbb{1}_Y(x) = \begin{cases} 1 & : x \in S_Y \\ 0 & : x \notin S_Y \end{cases}$$

对于偏序信号 $\langle i,j,k \rangle$ ，更新规则如下：

$$u_{id} \leftarrow u_{id} - \alpha_u(1 - \sigma(\Delta \hat{h}))(1 + v_{jd} - v_{kd})$$

$$v_{jd} \leftarrow v_{jd} - \alpha_v(1 - \sigma(\Delta \hat{h}))(1 + u_{id})$$

$$v_{kd} \leftarrow v_{kd} - \alpha_v(1 - \sigma(\Delta \hat{h}))(1 - u_{id})$$

经过一轮的偏序信号的更新，采用如下规则对聚类信号 $\langle i, j, k, l \rangle$ 进行更新：

$$u_{id} \leftarrow u_{id} - \beta_u(1 - \sigma(\Delta \hat{f}))(v_{jd}v_{kd} - v_{jd}v_{ld})$$

$$v_{jd} \leftarrow v_{jd} - \beta_v(1 - \sigma(\Delta \hat{f}))(u_{id}v_{kd} - v_{id}v_{ld})$$

$$v_{kd} \leftarrow v_{kd} - \beta_v(1 - \sigma(\Delta \hat{f}))(u_{id}v_{jd})$$

$$v_{ld} \leftarrow v_{ld} - \beta_v(1 - \sigma(\Delta \hat{f}))(-u_{id}v_{ld})$$

实际情况中，更新会一直持续到目标收敛或者到达最大更新次数。

3.6 实验

3.6.1 实验设计

我们采用的数据集来自Youku,它是中国最大的视频分享网站。为了获取足够密集的用户-视频反馈数据，我们从一段时间内最流行的视频开始抓取，从这些视频出发，我们抓取关于这些视频的所有评论，所有抓取操作都是通过Youku的API完成。根据80-20[?]原则，观看数量靠前的流行视频会覆盖大多数的用户。由于我们无法获取用户的观看记录，只能使用评论数据作为一种替代品。评论一般会在用户观看完视频后被发表，因此使用评论数据比较可靠，并且评论是比观看记录更强的一种正例反馈。

因为我们使用的是评论记录，稀疏性问题显得更加严重。具体而言，我们抓取了评论数不小于40的3130个视频，并得到10918个用户的数据，每个用户有至少10次评论历史。整个数据集的稀疏度是99.05%。值得注意的是，现在我们

将会对用户的评论行为作出预测，相比于观看行为，这会是一个更加艰巨的任务。

为了验证使用聚类信息的有效性，我们将和基于偏序的推荐模型BPR进行比较，它是我们模型中仅包含偏序信息的那部分。

3.6.2 参数设定

在推荐系统领域，人们关注的是推荐的成功率，即精确度。但是更细化的指标还会关注命中物品的排序，希望它们的排序越高越好。为了评价算法的有效性，我们使用如下的指标：

精确度 推荐列表中命中的视频占推荐总数的比例

召回率 命中的视频占用户观看过的所有视频的比率

平均准确率 精确度里考虑命中视频顺序的一个版本，信息检索中常用的一个指标

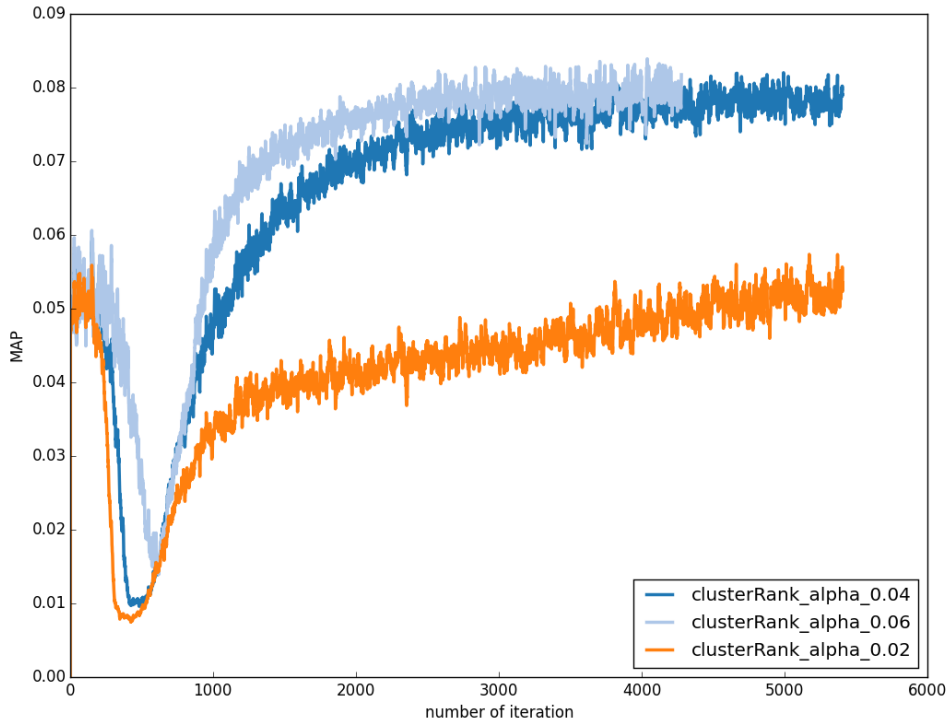
可调节的参数包括隐含因子的维度 D ，惩罚项的权重 α 和 β 。为简化实验我们设 $\alpha_u = \alpha_v$, $\beta_u = \beta_v$ 。为了得到最优结果，设定 $D = 45$ 。其中最重要的参数是惩罚项，对于聚类信号的惩罚项，是定 $\beta = 0.001$ 以尽量减少对聚类更新步骤的惩罚。

如图??所示，我们得到了随着 α 的改变得到的平均准确率的变化，横轴代表迭代次数，纵坐标代表平均准确率取值。因为我们使用随机梯度下降优化，结果中存在着上下波动的情况。但是长期来看，结果曲线还是呈现出上升的趋势。结果中的平均准确率先是下降，之后呈现稳定的上升趋势，直到收敛。当我们将 α 从0.04上升到0.06之后结果没有明显的改变，因此我们将设定 $\alpha = 0.06$ 。

3.6.3 结果分析

在图??中我们给出了对于top-10的推荐的效果，从左到右分别对应着精确度，召回率和平均精确度，两条曲线分别对应着BPR和我们算法的结果。对于精确度和召回率，我们只得到了微弱的提升：分别从0.08到0.09以及从0.04到0.045。但是对于平均准确率，结果的提升非常可观，从0.045上升到了0.075。

这个结果说明我们的算法并不会得到许多全新的命中视频，更多的实在原来的推荐列表中优化推荐排序，它的特点在于能够放大相关度高的视频的权重，

Figure 3.5: MAP for different α

使得它们在推荐列表中的次序更高。这点在视频推荐系统中十分关键，因为视频推荐所能提供的推荐空间非常有限。通常一次只能有 5 个视频能够在推荐列表中显示，对于移动设备的用户，情况会变得更糟糕。

如图??，我们看到对于top-5的推荐结果变得更加有趣。这一次，我们的方法在三个指标中的结果的表现都远胜过BPR的结果。相比top-5与top-10的结果，我们发现在准确率与召回率上的提升主要的原因是：原来在5-10名的那些命中视频被准确地推送到了top-5的列表中，这再次证明了我们的方法在推荐次序上优化的有效性。

我们观察到，当算法收敛之后，我们的方法在结果上表现得更加稳定，而BPR在结果上还存在比较大的波动。这再一次证实了仅凭偏序信号不足以得到精确的推荐模型，从结果上看出在搜索过程中算法还存在模棱两可的状态，然而聚类信号的引入使得这种模棱两可的状态被消除，算法能够很明确的知道哪个解是更优的，进而表现出的结果也更加稳定。

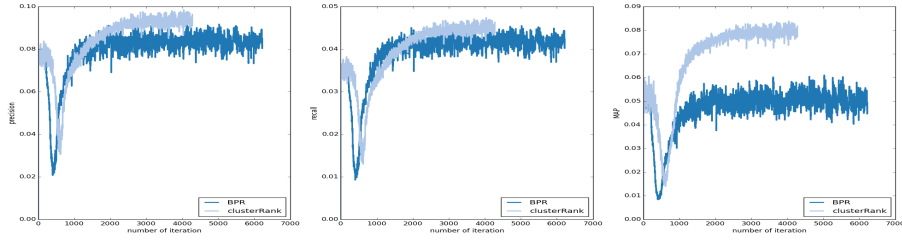


Figure 3.6: Performance for top-10 recommendation

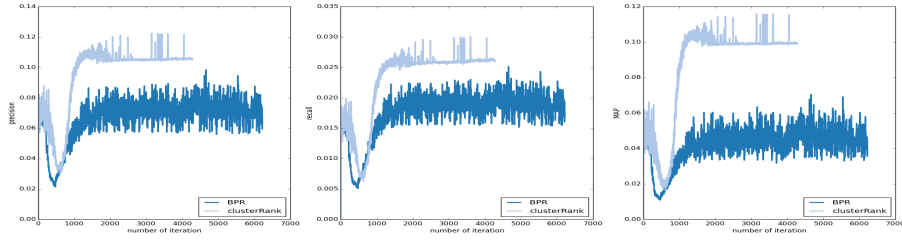


Figure 3.7: Performance for top-5 recommendation

3.7 本章小结

本章介绍了聚类融合的视频推荐算法。该算法融合了用户的观看聚类现象来减轻推荐中的“冷启动”问题。实验结果表明我们的方法能够有效地将相关的视频推荐到推荐列表前排，鉴于视频推荐一般仅提供有限的推荐空间，这些提升对于视频推荐有重要意义。我们的算法是跨领域学习的一个例子，不同点在于我们从一个数据源中得到不同类型的数据。这种做法适用于任何偏序关系和聚类现象并存的数据。

第四章 基于聚类推荐的Ad Hoc网络缓存机制

4.1 背景介绍

我们生活在一个在线视频疯涨的时代。主流的视频网站(如YouTube, Hulu, Youku)以及社交网站(如FaceBook, Twitter, Weibo)迅速地占据了互联网流量的大部分。每天,在很短的时间内都会有不计其数的新视频流入互联网。据悉,在线视频已经占领了世界互联网流量将近一半的份额(42%的流量来自流媒体播放)。世界最大的播放平台,YouTube,就已经消耗掉流媒体流量中57%的流量,这也就是说,单独YouTube一个视频应用就包揽了全球带宽的24%。

面对大量的视频流量,运营商想出各种办法来控制,其中最重要的方法就是缓存。规模大的内容提供商会在物理上的各个区域提供对应数量的缓存服务器,当来自各个区域的请求涌入服务器时,服务器会指导各个客户端向各个区域的缓存服务器索取内容。这么做能够极大的减少网络流量在骨干网络的拥塞,同时降低海量请求对中心服务器造成的压力。

通常运营商会根据历史请求的多少来选择是否缓存某个内容,如果某个视频是在某个区域内有大量增长的趋势,那么它就会被缓存。这涉及流量统计以及缓存替换策略的设计。然而,这种传统的做法需要在大量的请求出现之后才能制定相应的对策。如今的流媒体环境的更新速度已经极大的加快了,通常视频流行的周期快的能够在2天的时间之内结束,这些周期的特点是不可预测且短暂。基于统计数据的缓存策略在快速的社交媒体消费环境下容易散失其优势:在大量统计之后得出结论却发现请求已经不再产生。

应对这个问题的一种方法是主动缓存策略,它不依靠大量的历史请求来决定是否缓存一个内容。与此相反,在一个内容在某个区域中流行起来之前,就决定是否缓存一个内容,这样做能够很好应对未来将要产生的爆发式请求。但是,要使主动缓存达到效果,就必须能够预测某个视频在未来的流行程度,这一点涉及视频属性的分析和对产生请求的用户属性的分析。

随着移动设备的普及,人们开始习惯在手持移动设备上完成各类流媒体消费。在中国,3G与4G信号的全面广泛覆盖让人们每时每刻都能享受到互联网带来的便捷。但是,不同于有线网络通信的是,无线网络是一个开放式的媒介传输,当一个用户发送或者接受数据的同时,周围的物理媒介就会被占用,同样物理范围内的用户就必须等待其完成传输,否则就会造成信号的冲突。这就

相当于一个物理区域内的用户同属于一个总线网络，越多的请求就会造成越严重的阻塞与冲突。

另外，无线网络自身存在一个通行能力的上限。因此，想通过增加通信能力来减少冲突的做法不太可行。在人员密集的场所，如体育场，大型游乐园，城市商业街区等地会有大量用户同时争用一个信道，传统的基站通信在这种环境下的通信能力有限。于是，人们提出在传统的基站式的通信方式上增加设备之间的通信方式。这种通信方式形成的网络称为移动自主网络。

移动自主网通过蓝牙或者WiFi信号进行设备之间的通信，原本在基站和终端设备之间传输的数据通过移动自主网可以在设备与设备之间传输，典型的应用场景就是流媒体播放。通过构建移动自主网，用户发出的流媒体请求能够被导向到网络中存在该资源的设备上，通过移动自主网来进行视频的传播。通过控制设备发送信号功率等方法，设备之间传输所占用的信道的物理范围远远小于基站的覆盖范围，进而极大的提高该区域内无线网络的整体吞吐量。

4.2 Ad Hoc 网中的视频传输

虽然设备之间传输的模式能够减少干扰范围，但是其仍旧属于开放媒介传输方式，设备之间还是会存在相互干扰的情况出现。尤其对于流媒体这样的应用，信道的争用问题依然很严重。假设在一个自主网络中，每个用户都持续性的发出流媒体视频请求，那么信道就会始终处于被占用的状态。激烈的竞争导致冲突频繁出现，最终每个用户都得不到预期的带宽，视频质量进而下降。

可以看到，在无线网络中，保证通信顺畅的有效途径是减少信道的争用，尽量避免冲突的产生。结合到视频传输的具体应用场景中，那就是尽量减少用户传输视频所造成的信道占用。在满足相同数量请求的情况下，达到这一效果唯一做法就是广播与缓存。

广播是一种有效利用信道的方式。无线网络中，无论是广播还是单播，都会占用信道。如果能够利用广播覆盖更多的目标，就能够减少许多冗余的数据传输。缓存最初的目标在于通过减少播放延时和播放卡顿现象提升用户体验。在自主网环境中，设备可以通过监听信道中传输的数据，缓存自己认为需要的内容，进而达到降低传输开销的目的。

在本章，我们提出基于推荐的缓存机制。其中的考虑在于用户可以监听到自主网络中的数据，但是网络中的数据量庞大，而设备本身包含的存储空间无法容纳所有的数据，因此在缓存过程就要有所取舍，如何取舍就是推荐的任务。与传统的缓存任务不同的是，我们缓存下的内容有极高的可能性不会被利用到，

这个概率要取决于推荐的准确度。但是，相比于网络信道资源和用户的播放体验，我们认为设备的存储空间是廉价许多的资源，使用部分的存储空间换取网络信道资源和播放体验的提升是值得考虑的。

假设在一个100个用户的自主网络中，每个用户都在持续的发出视频请求，网络中存在500个视频资源，设备的缓存空间足够存储将近30个视频。由于用户发出的视频请求并非同步的，因此在不同的时刻，会有不同的用户向同一个视频发出请求，那么该视频在网络中传输次数就达到了10次。如果能够在第一次传输该视频的时候，将这个视频事先缓存到其他9个设备的缓存中，那么，网络就可以减少9次的视频传播，同时也能提升另外9个用户的体验。这样做是一种变相的同步机制，使得用户的请求能够在一次的广播中得到满足。

为了预测用户将来会播放哪些视频，可以借鉴推荐技术。推荐技术通过历史数据以及用户视频属性来对用户的偏好进行预测，从而保证被推荐的视频在将来以一定的概率被观看。缓存策略成功的关键是推荐的命中率的提高。传统的协同过滤方法在数据稀疏的时候表现不佳，因此，本章我们使用我们的基于聚类的推荐框架来实现缓存任务。具体而言，系统通过给各个节点提供推荐，进而告知移动设备未来应该缓存哪些视频。之后，系统通过广播被推荐的视频给设备提供视频传输，这些视频通过一次的广播传播到达各个目标设备缓存中。这些目标设备中的任意两个观看了该视频，就达到了节省传输带宽的目的。但是现实情况是，大部分的用户观看的内容呈现zipf分布，即80%以上观看的视频集中在20%以下的视频内容，少部分的内容会被大量的用户消费。因此，我们期望一次的传输能够被大量(而非2个)的用户观看，进而使得大量的传输带宽被节省。

4.3 推荐算法

在传统的缓存应用领域，十分明显的特征是被缓存数据块都因为应用特殊性有极高的概率被使用。例如，传统的流媒体视频，长达将近一个小时的视频需要分成多个数据块，缓存系统可以根据当前观看点迅速判断出应该缓存观看点之后的那些数据块。因为视频的连续性，这些数据块天然的就具有很高的概率被使用，因此缓存也有了一定的命中率保证。

不同于传统流媒体领域，现在互联网中的视频特征是时间短且类别杂，并且数量庞大，视频与视频之间没有时间上的连续性和先后顺序。那么，为了使被缓存的内容有一定的被使用的概率保证，我们需要整合统计数据，归纳出合理的模型来判断用户对视频的偏好程度，推荐算法正是这些场景下的解决方案。这里有两点需要注意：

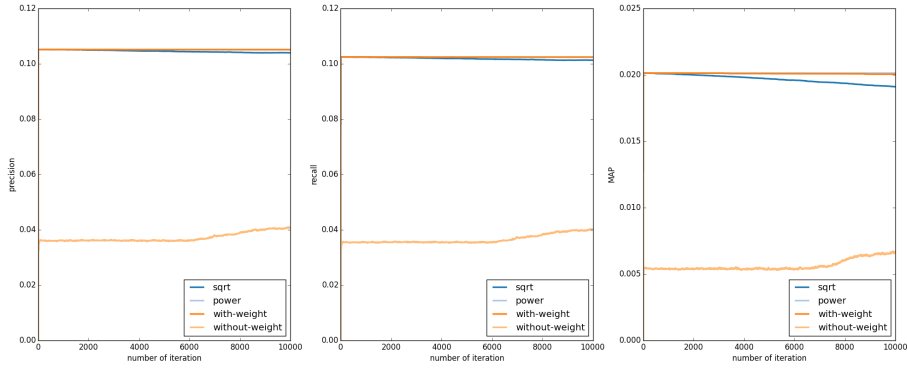


Figure 4.1: 加权vs 不加权效果对比

1. 传统流媒体缓存是纯粹的缓存，因为用户只有可能看接下来的视频片段，这其中根本没有推荐的必要，因为用户对于一个视频内的片段没有选择的余地。短视频流行的应用场景下，缓存可以是其中一个目的，但是更多的是推荐的成分，它的最终结果可以与纯粹缓存一样：减少时延与抖动，减少传输代价。不同点在于，推荐可以影响用户行为，而不论缓存内容为何，用户的观看行为都不会改变。在本章，缓存和推荐的意思都一样，即为了达到缓存的目的而使用推荐技术的行为。
2. 缓存命中率的差别。传统缓存一般都有很高的命中率，对于正常用户而言，命中率可以达到接近100%。然而推荐的精准度却远没有达到这种水平，在我们的数据集中，精准度最高只能达到16%，而对于稀疏的数据可能10%都不到。因此我们提出使用广播的机制，我们最终的目的不在于精准度，而是在于传输成本的节省。采用广播机制，一次广播可以达到多次缓存命中的效果。因此，这里是使用存储换带宽的一种权衡。当然，即使如此还是必须保证一定的命中率，毕竟存储空间也是有限的。

4.3.1 推荐模型

我们要使用的推荐模型是第三章提出的聚类推荐算法，虽然在大数据集合下它表现出了优势，但是在小用户群体中的结果还是个未知数。具体的，我们考虑只有1000个用户和500部视频的场所。

对于大型的CDN应用场景，根据流行度来决定缓存内容是常见的策略，而且一般都是有效的，这里我们将其作为baseline与我们的方法进行比较。同样的，我们也比较原始的BPR算法，在数据密集场合，这或许能够表现得更好。

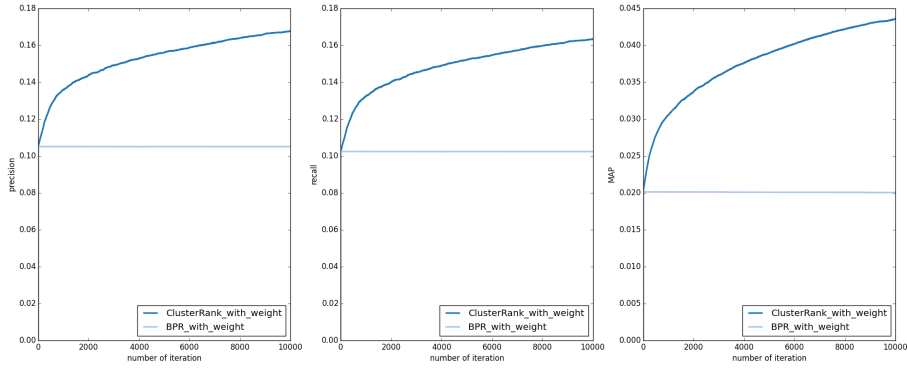


Figure 4.2: 加权情况下的效果对比

在小数据集中，BPR的表现有很大的不同。具体而言，在我们的算法中，我们会对每个用户都进行一次更新，而BPR在这样的更新下表现得一般，甚至不如稀疏场景下的结果，这一点并不在预期之中。于是，我们进一步对BPR进行改进，考虑进时效性，对每个用户最近的观看记录给予更高的学习权重，离当前时刻越久权重也就越低，我们比较了多种加权策略，效果如图??所示，在使用加权策略之后，精准率有了明显的提高。

为了获取一个更好的结果，我们的聚类算法也建立在加权偏序更新的策略之上。然而，对于聚类信号的更新，我们维持原来的更新方法。因为聚类信号更新是一种基于相似性的计算方法，相似的视频对每一个用户而言在任何时刻都是相似的，因此就无所谓是否加权。

在使用加权策略之后，聚类信号带来的另外一个好处是为加权策略提供一个更加平滑合理的加权分配。例如，原本的加权策略是根据排名进行加权，对最近的视频给予1的权重，次近的视频给予 $\frac{1}{2}$ 的权重，依次类推。但是，这种策略并不完美，例如，假设一个用户最近连续观看了2个视频，它们在时间上靠得非常近，那么其中一个视频的权重是另外一个视频权重的2倍就显得很不合理。为了弥补这个问题，聚类信号会将这两个视频在特征上往相向的方向逼近。并且，我们的聚类信号正好是基于时间分簇的，因此聚类更新只会针对实际上相近的视频进行更新。这样，聚类更新就会对不合理的权重进行调节，将原本几何衰减的权重调整至符合实际情况。图??给出了结果对比，与预期一样，聚类信号的引入同样提升了精准度，越高的精准度为我们的缓存策略的有效性提供了更大的保障。

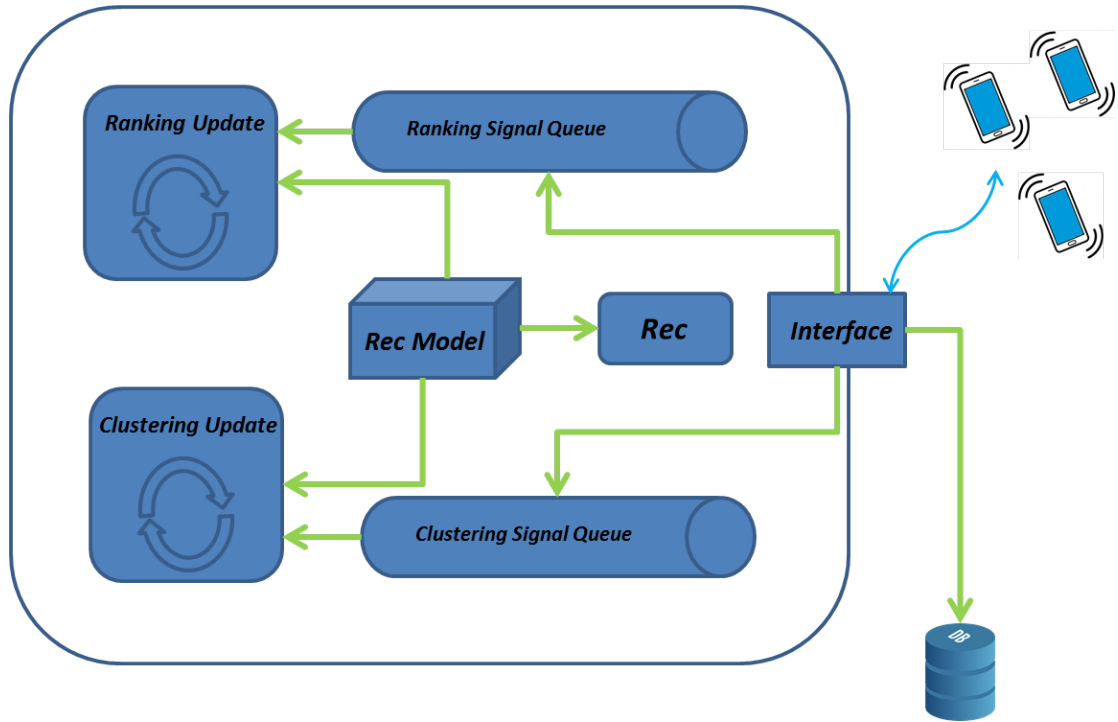


Figure 4.3: 推荐预取系统框架图

4.4 系统设计

推荐算法分为离线算法与在线算法，在实际系统中，在线算法比较实用。原来的聚类推荐算法是在离线情况下训练完成，欲将其改为在线算法，必须重新考虑更新策略。

原来的更新过程是，实验平台提供一系列数据，优化算法遍历数据集中的每条数据，完成更新步骤。为了使得模型达到收敛状态，更新过程不断循环进行。为了搭建一个推荐框架，系统中存在两类实体，客户端和服务端。客户端负责提出推荐请求，将自己的使用数据发送到服务器，同时完成实际的缓存工作(监听网络中的视频内容，选择被推荐的视频内容进入缓存)；服务端完成实际的推荐工作，具体而言，它负责记录客户端的使用记录，定期的进行模型的更新，当用户发送推荐请求之后发送推荐列表，另外，为了使广播达到带宽节省的效果，服务器还负责对推荐视频进行传输上的同步，即让其在一次的广播中完成传输。

图??是系统的架构图，它包括交互接口模块，数据缓存队列，模型更新模块以及推荐模块。交互接口模块负责将用户行为日志转换成偏序记录和聚类记录分别存入偏序缓存队列和聚类缓存队列，同时将原始的历史记录存入数据库；

数据缓存队列分为偏序队列和聚类队列，因为是先进先出数据结构，它保证了队列中始终维持着最新的用户历史记录。队列的容量需要被谨慎的选择，当系统中用户数量增加或者视频数量增加，都要求容量对应的扩展，因此系统会记录下当前系统的用户和视频数量，队列的大小作出对应的动态调整。同时，队列中的内容也需要被监控，用户时常会在一段时间内迅速发出大量请求，这样队列中的数据将充斥着这群用户的数据，更新出的模型会向这群用户偏袒。为了使得不同用户得到平等地更新，必须控制每个用户的数据在队列中的数量，同时跟踪每个用户的记录在队列中的位置，超过阈值的用户的数据将优先被请出队列。

模型更新模块也分为偏序信号更新模块和聚类信号更新模块，一次更新迭代需要对每一个训练数据完成一次更新，为了使模型更新完成，通常需要多轮的更新迭代。这也是我们需要维护一个数据队列缓存在内存中的原因，数据通常需要被访问多次之后才能充分挖掘出其中的信息，将数据保存在内存队列中，保证了其在队列中的时间内被充分的更新。但是，当数据更新频繁，数据很快就会被请出队列，但是这正是我们所期望的，这保证了：当请求频繁的时候，我们针对最新的请求进行更新，当请求速度缓慢之后，我们能够有足够的请求用来更新模型(即一个数据被多次迭代更新)。另外一个关键点是模型更新的频率，由于采用随机梯度下降进行更新，一次迭代需要的时间很短，模型更新频率要视请求到来的速度决定，当请求多的时候，模型更新频率要相应提高，请求速度平静之后，频率可适当下降。

推荐模块针对每个人的特征向量，给每个视频计算一个偏好值，根据偏好值排序推荐前N个视频。实际实现出可使用一个固定大小的优先队列快速获取前N个视频。为了保证数据传输的同步性，推荐模块被周期运行，因此每个优先队列也周期性地被更新。如果进一步考虑视频的时效性，即给新视频更高的权重，那么在计算权重时候可适当提高新视频的权值而非降低老视频的权值。在每一个周期里，推荐模块向各个客户端发布推荐列表，受到多数确认之后知道视频服务器对所有被推荐视频进行广播发送，各个移动设备根据自己的推荐列表进行接收。

客户端模块的行为很简单，只要及时将观看数据发送到服务器，同时接收推荐列表以及根据推荐列表监听缓存对应的广播数据。一个历史数据的生命周期如下：首先由客户端产生，而后被传输到服务器接口，一份保存于数据库中，另外一份转换成偏序信号和聚类信号，在对应的队列中被更新模块中多次更新之后被请出队列，进而结束其生命周期。

4.5 模拟实验

移动网络日益普及的今天，人们开始像使用台式设备一般使用移动设备；同样的，人们开始像使用有限网络那样开始使用无线网络，无线网络带来的便捷性却是有线网络所不能比拟的。即时聊天，收发邮件，浏览网页甚至是观看视频，很多曾经仅在台式设备上出现的应用现在已经普及移动应用市场，在不久的将来，视频流将称为移动通信的主要瓶颈。

之前提到无线通行存在固有的通信瓶颈，提高无线网络吞吐量的有效途径是建立设备间的无线通信。然而，无线通信要保证信道足够顺畅以尽量避免冲突产生，为此广播通信成为大范围内容传播的首选

4.5.1 实验场景设定

为了验证推荐预取带来的实际效果，我们进行了模拟实验。实验场景假设用户数量固定且视频数量固定，这些用户在区域内组成无线自主网，同时每个用户都在4G基站的覆盖范围内。用户根据自己的情况不断发出视频请求(不同用户发送请求的频率可能不同)，请求发出后，无线自组网的资源定位模块定位相应的资源，引导该节点向请求节点发送视频，视频在网络中传播的同时能够被其他节点监听，它们有机会将其缓存到自己的设备上。

通常，无线自组网络中的节点发送请求的方式是任意的。例如，节点1发送视频A的请求，过了一段时间后，节点2同样发送了视频A的请求。如此，视频A就在网络中传播了2次，对于热门视频，由于请求的不同步，视频会被重复多次在网络中传播。为了减少视频A在网络中的传播，一种方法就是采取缓存策略。当节点2监听到视频A的传输时，将其缓存。但是，节点2能够监听到网络各种视频内容，数量远超过它的缓存空间大小。因此节点2必须决定缓存哪些视频，具体而言，当视频A来到的时候，它不知道将来视频A是否有用。为了判断视频A对节点2的价值，我们使用推荐技术来给视频A作出评价。具体而言，在所有现有缓存中，利用推荐技术淘汰掉价值最低的视频，将A缓存下来，如果A的价值最低，直接忽略A。

本实验通过测量用户接受缓存内容与实际请求所占的比例，来计算推荐缓存为网络减少了多少视频流量。例如，我们假设每个用户未来都还要观看5个视频，我们通过推荐缓存，为每个用户缓存25个视频，如果一个用户观看了缓存中的其中一个视频，那么认为原本需要网络传输的一个视频由缓存代替，由此节省了一个视频的传输。这里作出的权衡是：网络带宽和缓存空间的取舍。在上面的例子中，我们认为一个视频的传输的代价要高于5个视频的缓存，5个

视频的缓存减少了一个视频的传输和观看实验以及各种视频卡顿现象，在存储设备日益廉价的今天，这种取舍还是合理的。

4.5.2 实验数据

这里使用的数据仍然是Youku的数据，数据集中提取最密集的一部分用户和相应的视频。虽然数据中每个用户的请求时间不同且间隙不同，我们假设这些请求都集中在实验假设的时间段内，并且每个用户的请求进度相同(即相同时间请求视频所占自己所有请求的百分比相同)，视频请求的顺序维持原来的顺序。对于缓存命中，我们做了一个假设，如果一个缓存是该用户未来观看的视频，即使不是下一个要观看的视频，我们仍然视其为命中。

由于缺乏现实用户的请求模式，上面我们假设了一个同步的用户请求模式，为了简化实验步骤，我们对视频缓存步骤和实际请求步骤也进行了同步，即所有用户统一进行缓存步骤，当所有缓存结束之后，所有的用户才开始实际的请求步骤，当请求了一定比例的视频后，新一轮缓存才再度开始。我们选取了1000名用户和500个视频，将其置于假象的Ad Hoc网中，我们做了如下假设：

1. 如上所述，请求和缓存受到了同步
2. 用户之间传输需要一个单位的代价，广播一次也是一个单位的代价
3. 每个用户在请求期间会观看10部视频
4. 只要缓存的视频用户将来实际管看过，就代表一次缓存命中，且缓存命中可以减少一次请求消耗

4.5.3 评价标准

基于上述假设，我们做了三组实验，分别对应着改变缓存大小，该表网络用户数量和对比不同推荐算法在效果上的差别。实验中，我们提取每个用户前50%的观看记录作为历史记录，训练过的模型用于模拟缓存，时间轴上的后50%历史记录作为命中的标准。实验是为了得到每种情况下节省的通信带宽，其计算公式为

$$communication_save = \frac{(hit_num - distinct_videos)}{10 \times user_num} \quad (4.1)$$

其中的 hit_num 代表所有用户的缓存命中的所有视频，而 $distinct_videos$ 是推荐的不同视频的个数，由于推荐的视频通过广播传输，每个视频只需要消耗一个

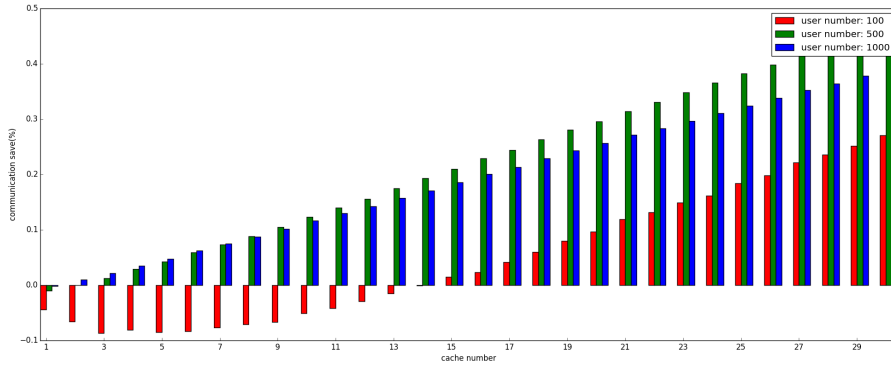


Figure 4.4: 传输节省随缓存数量变化

单位代价。 $user_num$ 是所有用户的数量，乘以10是因为我们假设每个用户将会观看10部视频。因此， $communication_save$ 就代表所减少的通信占原本通信量的百分比。

4.5.4 结果：改变缓存大小

客户端的缓存越大，保存的视频命中的个数越多，在假定用户观看数量一定的情况下，相应的节省的带宽也会越大。通常，缓存大小会远大于用户观看的视频数量，这个假设是基于移动场景下流媒体传输所固有的高成本，节省的一次传输开销，不但能增强用户体验，还能减少传输带来的信道占用。过多的信道争夺造成的不仅是传输的减慢，更能最终使得信道拥塞不可用，带来的影响是群体性的。然而，缓存空间是相对廉价的资源，通常并不会成为应用的瓶颈。移动设备的流媒体主流的以短视频为主，并且分辨率要求低，典型的一个视频大小在在40M左右，缓存25个这样的视频占用1G左右的存储空间，这对于现代移动设备而言并不是一个问题。

图??给出了传输节省伴随着缓存大小改变的结果图，其中包含100,500和1000个用户的网络的结果。总体趋势是缓存越大节省的百分比越高。当用户数量较少且缓存数量小的时候，缓存技术反而增加了网络的负担，这种负担是由于推荐的视频需要通过广播来传输，命中的视频数量还不足以抵消广播传输带来的开销。

观察不同用户规模之间的结果比较，我们发现100名用户的结果最差.这符合直觉，因为越多的用户就代表越多的命中，这样就有更多的用户得到广播带来的好处。但是，我们发现1000名用户的结果是低于500用户的场景的，这是因为这500名用户的历史数据较多，命中率高，而1000名用户使得通信基数变大，

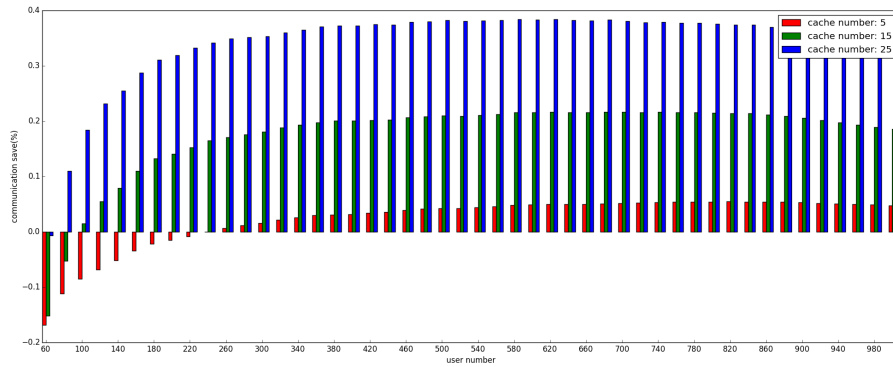


Figure 4.5: 传输节省随用户数量改变

为了维持百分比就必须保证一定的命中率，而另外的500名用户没法维持命中率。

从图中看出当缓存大于25能够达到30%的传输节省，这是基于每个用户观看10个视频的结果。考虑500人的场景，那么这就是5000次的传输，那么实际上我们节省了1500次的视频传输。如果实际上用户平均未观看这么多的视频，那么这个百分比会更高。

4.5.5 结果：改变用户数量大小

用户数量决定了广播传输所带来的效果如何，当用户数量不多的时候，就要求推荐缓存的命中率十分高，以至于能够抵消广播传输的代价。由于推荐一般属于个性化推荐，这意味不同的用户推荐列表会十分不同，如此一来广播传输的视频数量就会很多。因此，一般希望用户个数足够。

图??给出了缓存数为5,15和25的结果，当缓存数量为5的时候，通信节省百分比十分低但却十分稳定，这代表命中个数正在与用户数量等比例增长。从缓存数量从5到25的提升过程可以发现比例增长有所增快，因为在推荐列表顶部的视频一般极度个性化，所带来的增益并不明显。当缓存增加，列表尾部被缓存之后，更大众的视频被命中，从而带来的增益有所增加。当用户增加到一定数量之后，效果增加就不明显了，到300个用户时候，缓存的效果基本达到饱和状态。这里更多的看到是缓存数量的重要性，越多的缓存代表越广的适用场景和更高的传输节省，实际系统是希望能够处理各种用户数量场合的，因此设定更多的缓存很有必要。

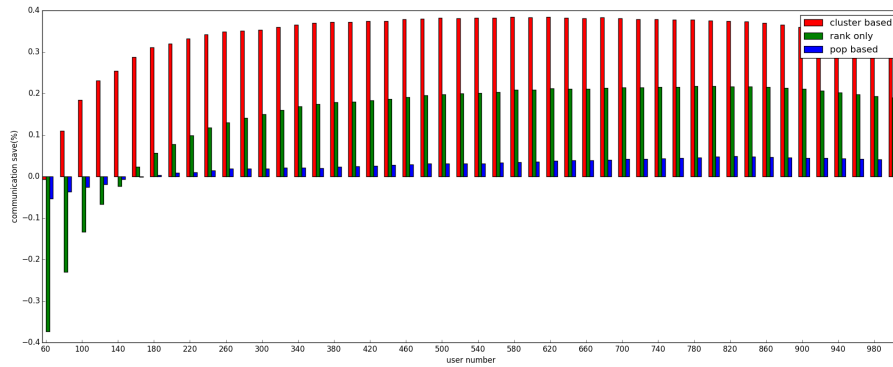


Figure 4.6: 不同方法的比较

4.5.6 结果：不同方法比较

命中率对于缓存而言是最关键的指标，不能保证命中率的缓存都是对带宽和存储空间的浪费。我们比较不同方法的效果，具体而言是

1. 基于聚类的推荐缓存
2. 贝叶斯排序推荐的缓存策略
3. 基于流行度的缓存

这三类方法分别对应着不同的命中率，其中1对比2将命中率从10%提升到15%。

首先观察基于流行度的缓存方法，这种策略将观看次数最多的那些视频缓存。虽然广播代价低(只需要传输63个不同的视频),但是极低的命中率使其无法带来很大的收益。尽管用户观看的视频呈现zipf分布，但是细化到一个有限的用户团体时候，用户之间的差异还是比较大，基于流行度的方式很难精准预测用户的行为。

基于聚类的推荐缓存策略是结果最好方法，相较于贝叶斯排序推荐(BPR),它拥有更高的命中率。但是除了关注命中率，推荐种类的多样性也值得关注。在缓存的场合，我们希望在保证命中率的前提下，尽量降低视频的多样性。这也是聚类推荐部分达到的效果，从结果中来看，BPR推荐的视频种类是494，而聚类推荐的视频种类是334，这为广播传输节约了开销。BPR过度强调个性化，以至于将很多不能确定的视频引进推荐列表，这些视频的命中率却很低，而聚类推荐能够很确定的将这些视频排除在外。这并不是说聚类推荐缺乏推荐结果的多样性，只是在选择上更加严谨，不会轻易为了多样性引入不相关的视频。

4.6 本章小结

我们提出将推荐方法应用到移动流媒体的缓存领域，在用户多样性复杂的今天，传统基于流行度的缓存策略已经不适用，推荐方法的引入极大的保证了缓存的命中率。我们利用无线信道的广播传输来减少单播传输带来的开销。模拟实验表明，推荐算法的引入能够有效的提高效果。另外，广播策略的有效性建立在一定的用户数量之上，缓存数量对结果影响十分明显。因此，实际应用中应该重视缓存命中率和缓存数量两方面。

第五章 总结与展望

简历与科研成果

基本情况

刘畅，男，汉族，1990年8月出生，天津人。

教育背景

2013.9~2016.6	南京大学计算机科学与技术系	硕士
2009.9~2013.6	南京大学计算机科学与技术系	本科

攻读硕士学位期间完成的学术成果

- [1] Chang Liu, Lei Xie, Chuyu Wang, Jie Wu and Sanglu Lu. “FootStep-tracker: an anchor-free indoor localization system via sensing foot steps“,in Proc. of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers.
- [2] 谢磊，刘畅，王楚豫，陆桑璐，”基于脚步感知的室内定位系统及其定位方法”，专利，申请号：201510373829.0
- [3] 谢磊，刘畅，陆桑璐，”一种基于交通流感知的智能交通灯调度系统及其调度方法”，专利，申请号：201410162206.4

致 谢