南京大学

研 究 生 毕 业 论 文

（申请硕士学位）

论　文　题　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　研究背景

视频应用作为互联网世界里的杀手级应用，一直主导着互联网流量的占用率．传统的视频传输被视为简单的文件传输，用户只有将所有的文件数据下载完毕之后，才能进行视频的播放．经过多年的发展，传统的视频传输已经被流式传输所代替。这种传输方法的特点是优先下载一小块亟待播放的视频数据存在缓冲区之中,当缓冲足够数据之后立刻开始视频的播放，这么做使得所需的存储空间大量减少，并且给用户提供了选择视频内容的机会。由此衍生出了两类的视频应用：

- 视频点播服务：这类视频服务的特点是允许用户在可选视频中选择喜欢的视频内容，流传输允许一定延时，但是视频选择空间大。
- 视频直播服务：视频的播放有着严格的时间限制，超过播放时间的数据片作废无效，关注传输的延时。

上述这两类视频服务统称为流媒体服务,它已经渗透到了我们日常生活的每个角落。传统的电视服务开始使用流媒体来提供直播服务；影片在线租赁平台也在流媒体技术之上将租赁业务扩展到更广的范围；社交网络中传播的主流内容使用流媒体技术；在线教育平台使用流媒体技术分享着稀缺的教育资源；广告商也发现，只有视频内容才能够吸引观众的眼球，用户的注意力不经意间就被转移到流式传输的广告之上。流媒体服务为互联网带来丰富多彩内容的同时，同时也给互联网通信能力提出巨大的挑战。尽管我们已经知道，在峰值时刻Netflix将会占据互联网流量1/3之一的比例，对于YouTube而言，这个数字还会上升到1/2.然而，这些还只是冰山一角。 5 年之内，80%的互联网带宽将被视频内容所占据。根据最新的报告[XXXCisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper], 到2019年，整个互联网将会变成一个巨大的视频管道。这些增长的部分原因是持续增长的互联网用户，不久的将来，世界人口的一半以上都将连接进入互联网；另一个原因是，网络上的每个用户将会消费更多的视频内容，视频质量也将会越来越高；这些都将给本已经接近饱和的网络环境提出更艰巨的挑战。

另一方面，视频服务的重心正逐渐从桌面平台向移动平台转移。移动设备从最初的只能提供通话和短信服务的终端逐渐进化成了具备高性能处理器，充

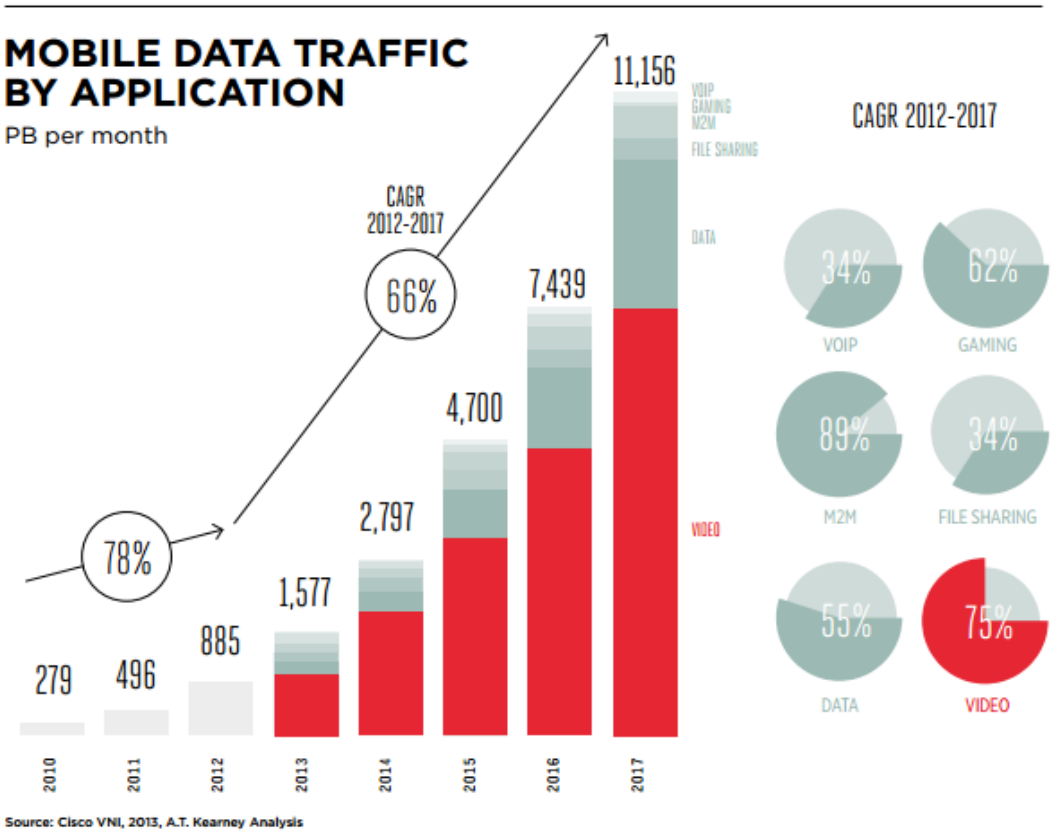

Figure 1.1: 移动应用流量增长趋势

足存储空间的移动计算机，人们已经习惯于在手持设备上浏览当日新闻，在社交网站上观看朋友最新的动态，或者在视频网站上看一部电影打发时间；这一趋势的最终结果是：移动用户的数量以飞快的速度增长，移动视频流量将会逐渐主导互联网，如图1.1。伴随着移动用户的增长，更多的视频将会由普通用户上传分享，视频内容也朝着更多样化的方向发展，如何从众多的视频内容中选择喜欢的视频，为用户提供良好的个性化的服务成为流媒体服务中新的重点。而底层传输技术也有了长足的进步，各种无线传输技术如蜂窝网，无线局域网和宽带无线网，使得如今的用户拥有了无处不在的连网服务。但是，无线网络在传输能力上有不可逾越的上限，网络速度的提升已经快要走到尽头，如何在移动用户爆炸的今天提供更好的视频传输，是进一步移动流媒体服务所不可回避的问题。

1.2　移动流媒体技术

流媒体是指在数据网络上按时间先后次序传输和播放的连续音/视频数据流，在传输的过程中只将部分的内容缓存，数据边传输边播放，这个过程中节

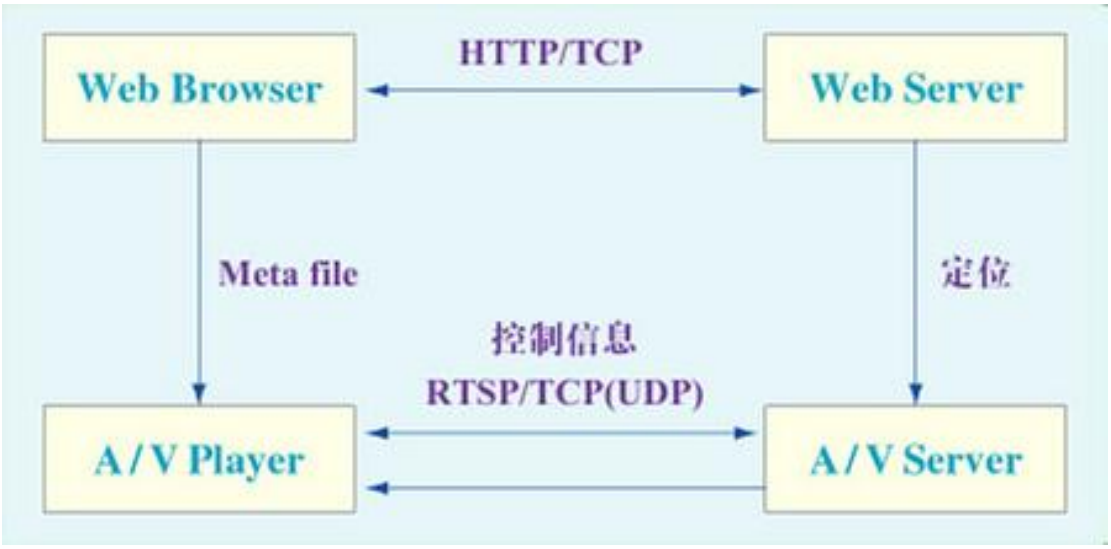

Figure 1.2: 流媒体传输过程

省下了传输的时间和存储空间。因为Internet 是以包传输为基础的异步传输，不同的包经过不同的路由到达目的端的时间会错乱，因此流媒体的传输需要缓存的支持，视频内容在缓存空间进一步按时间顺序整理。

流媒体视频传输的一般过程是：客户端在web服务器上寻找想要的资源，将请求发送至web服务器，web服务器将请求转发到相应的流媒体服务器；在客户端和服务器交换了控制信息之后，视频数据的传输正式的在客户端和流媒体服务器之间开始。经典的流媒体系统架构包含如下形式：

- 客户端/服务器架构：该架构模是传统的互联网应用的服务模式，请求和服务在一条链路上得到满足，客户端发送请求的目的地与视频内容的发送方一般处于一个物理区域。这种模式是所有流媒体服务的基础模式。

- 内容传输网络(CDN)：在很多应用场合，所传输的内容可以被策略性地放置到离客户端更近的物理区域，这些视频内容一般是整个网络中比较流行的视频，一次缓存必须能够满足大量的请求，视频内容的放置策略根据用户的请求模式而制定。

- 点对点传输(P2P)：随着请求用户的增加，服务器端的带宽需求也呈线性的增长，这对于用户基数庞大的大型系统而言是很大的一笔开销。点对点传输的特点是每个客户端同时兼具客户端和服务器的身份，在请求数据的同时也提供视频传输服务，而服务器只是起到协调的作用。

不同于有线网络中的视频传输，移动设备有着多种多样的屏幕大小，移动

设备所处的网络环境随着时间会迅速的变化。为此ABR被提出来，在不同网络容量情况下动态地调节视频的码率；同时SVC技术使用分层内容传输方式，随着更多的数据到来，更高质量的视频内容会被合成。在移动流媒体传输的末端，视频内容通过无线网络传输到客户端。这段传输过程是另一个值得关注的方面，在未来移动流媒体的传输瓶颈很可能就在这一环节产生，无线传输包含如下思路：

- 基站-设备传输模式：这是传统的无线传输方式，在基站覆盖范围内的所有移动设备分时共享一个传输频带。因此，要保证一定的视频传输速率，就要求传输速度够高，并且保证没有太多的用户共享传输链路。

- 设备-设备传输模式：在基站传输模式中，用户的可用带宽随着用户数线性减少。因此这种模式无法满足大用户量的区域的视频传输需求。设备之间传输的思路是使用不同的频带完成设备之间传输内容的任务，通过控制信号强度保证一定距离外的用户之间不会相互影响。

随着社交网络的流行，更多用户倾向于自己上传有趣视频，同时好友之间的这类视频得到更多地分享。在社交网络中传播的视频的特点是长度短，数量和种类却十分庞大，这改变了传统流式传输中所拥有的时间连续性，用户的观看行为被碎片化。这些使得为用户提供个性化的流媒体服务变得日趋重要，诸如缓存等重要的流媒体技术在新的应用场景中需要更多的了解用户视频之间的交互规律。所有这些都要求在凌乱的历史记录中获取用户的消费规律，为提升服务质量做充足的准备。

1.3　主要技术挑战

相较于传统的流媒体服务，移动流媒体因其特殊性，给流媒体服务设计带来了新的技术挑战：

- 用户的手持设备可以随时录制想要分享的视频内容，如今的视频集合中，更多的是用户生成内容。大量的视频内容充斥在视频分享网站里，寻找高质量而又感兴趣的视频变得异常的困难。因此，绝大多数视频如今都提供视频推荐服务，这不仅仅单纯为了满足用户的需求，更有利于视频网站的内容部署。研究表明，用户的观看行为很大程度上受到推荐的影响，大量的视频点击是来源于对用户所推荐的列表。

- 使用移动设备观看流媒体视频已经成为一种趋势，不久的将来无线设备就会面临来自大量用户的视频请求。届时无线网络技术能否支撑住大量的视

频数据传输就成为一个严重问题。流媒体服务长久以来一直关注着服务器端的优化，在移动应用场景中，无线传输环节才是真正的瓶颈所在。在变化的无线网络环境中，用户的播放过程会伴随着长时间的启动过程，播放过程的抖动，视频质量下降以及视频播放卡顿停止等问题。

对于第一个问题，推荐系统提供了一个有效的解决方案。多数现存的推荐系统使用用户的行为数据进行推荐，常用的信息包括用户的点击和评分，此类技术称为协同过滤。除此之外，用户和视频的一些描述性数据也被推荐系统使用，此时根据用户和视频在内容或描述上的相似性进行推荐，这类技术称为基于内容的推荐。协同过滤使用用户观看过的视频来挖掘它们之间的相似性，然而一个问题是，一个用户只能对有限的视频内容进行评分，而且系统中不断有新的用户和新的视频加入，这种数据稀疏和冷启动问题经常制约着这类算法达到预期的效果。基于内容的推荐方法虽然能够有效解决上述问题，但是用户和视频必须经过复杂的特征识别和提取过程，在现如今的流媒体系统中，海量的用户和视频数量让这个过程成为了基本不可能的任务。

对于第二个问题，更多的优化侧重于客户端优化。例如移动设备可以使用ABR技术动态调节的码率接收数据，或者使用SVC技术分层传输视频内容，保证能够传输低质量的画面，之后在原始层上进行加强。然而，这些优化仅仅关注单个客户端，在基站分时传输模式下，一个基站只能服务于有限的用户数量。首先，无线传输能力有一个上界，那么想通过增加传输速率来服务更多用户已经不可能；另一方面，视频的质量变得越来越高，对更高传输速率的要求已经不可避免，届时可能的策略只有限制视频传输，或者任由流媒体服务质量下降，这都是服务提供者和用户所不愿看到的。如果同时提供大量用户在无线场景中使用流媒体服务是一个亟待解决的任务。

1.4　主要研究内容和工作

为了应对持续增长的视频数量，一个良好的视频推荐系统对于视频服务而言是不可缺少的部分。然而，如今的推荐模型或多或少受到数据稀疏和冷启动问题的困扰，而视频应用正是数据稀疏和冷启动问题十分严重的应用领域。为了提高推荐的效果，推荐模型需要能够利用到各种可用的信息进行推荐，并且能够结合具体的应用场景获取特殊的有用信息。

同时，突破无线网络传输限制的一个方案，就是使用设备之间传输。其基本思路是通过限制传输距离做到设备之间的传输互补干扰，从而达到频带空间上的复用。然而，即使对于设备间的传输而言，还是存在带宽上的限制。所以，

对于设备之间传输的良好规划还是必不可少的。

基于上述目的，本文提出了一个基于聚类信息的推荐算法；在这个推荐算法之上，建立了一个基于推荐模型的缓存模块；具体工作包括如下两点：

- 基于聚类信息的视频推荐模型

  该算法基于隐含特征建模，给每个用户和视频赋予一个隐含特征向量，代表偏好值在不同因子上的强弱。传统的推荐模型使用单纯的一类信息，即偏好信息。通过构建用户对不同视频的偏好区别，来训练用户和视频的特征向量。我们发现，在视频应用中存在着独有的聚类现象，即用户在一段时间内所观看的视频的类别十分相似，并且对用户的偏好而言也是相似的。我们利用这种独特的相似性的聚类，从这类视频观看在时间上的聚集现象中提取关键信息，用来加强用户和视频的隐含特征的训练。这么做达到两个效果，一是直观上提供了对特征向量更好的解释，它能解释视频属性，使得用户同一时刻观看的内容对于该用户而言是相似的；二是从它侧面上增加了可用数据的数量，部分解决了数据稀疏性和冷启动的问题。实验结果表明，通过增加聚类信息对推荐的各方面的指标都有明显的提高。

- 基于聚类推荐的缓存技术

  一种提升无线流媒体服务的方法是使用无线自组网，在此之上，进一步利用无线传输的广播特性，移动设备可以缓存“未来观看的”视频内容。这其中的关键问题是，难以判断哪些视频是用户未来将会观看的。利用推荐系统能够改变用户观看行为这一特点，我们在无线自组网流媒体系统中添加了一个推荐模块，该模块用来对移动设备的监听缓存做出决策，即决定应该缓存哪些内容，以及应该淘汰缓存空间中的哪些内容。通过监听缓存方式，并且在一定缓存命中率保证的前提下，整个系统的吞吐率能够得到良好提升。模拟实验结果表明，在移动自组网中使用基于聚类推荐的缓存策略，并且在一定用户数量保证的基础上，网络中所需传输的视频数量能够被大量地减少。大多数的请求都被监听到的视频内容所满足，因此一次传输能够达到满足多个请求的效果。

1.5　本文组织结构

1. 第一章节介绍相关研究背景，移动流媒体的主要服务模式与架构，其中存在的种种挑战，并对本文的组织结构进行了介绍.

2. 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

3. 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.

4. 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

在移动场景中，系统的设计和问题的关键已经远远不同于有线场景，移动特性使得问题增加了新的复杂度：

- 用户随时随地分享视频内容的能力使得视频的库存爆炸式增长。不同于传统流媒体应用中，公布的视频是运营商精心选择和处理的优质内容，用户不需要花费大量时间进行筛除和选择。如今视频库内混杂着大量用户视频，如何帮助用户寻找高质量且感兴趣的视频是一个复杂的问题。
- 无线网络的能力限制使得系统设计者必须寻找别的方式来提高视频传输能力，缓存作为一种有效的技术手段解决了大量的传输方面的问题。然而，在缓存系统的设计上，还有很大的发挥空间。

本章介绍本文的相关工作。首先介绍移动流媒体的架构和服务模式；接着介绍流媒体个性化服务相关的内容，由于个性化服务与推荐系统高度相关，重点介绍推荐系统的工作；最后介绍现有的缓存的技术和工作。

2.1　移动流媒体服务

在几十年的流媒体发展过程中，技术和架构模式发生着巨大的改变，流媒体包含如下架构模式：

- 客户端/服务器模式(C/S)：在1990年到2000年之间，研究的重点集中在新的流式协议的设计和实现之上，例如针对流媒体传输的实时传输协议(RTP)。最初的流式传输协议被整合在多媒体播放器之内，这些播放器直接经过因特网从流式服务器获取视频流。随后，人们很快发现通过标准的网页浏览器来传输视频流更为方便，因此，提出了使用HTTP来传输数据块的思路。由于标准HTTP协议的便捷性，HTTP流传输很快被工业界所广泛采用。[XXX CS模式图，HTTP传输]
- 内容传输网络(CDN)：C/S架构中的客户端和服务器的链路可长可短，对于长距离链路，最好将大数据块传输路径长度缩小，CDN即是实现这一功能的技术。CDN是配置在网络边缘的一系列内容服务器所组成的网络，其与生俱来的分布式特性让它拥有处理持续增长的内容需求的能力。大型

系统中的流行应用和内容被放置于离终端用户尽可能近的地方，将响应延时大量减少，进而提升了服务质量。为了限制服务器负载，处理资源匮乏，或者应用转移，在CDN中有相应的负载均衡策略提供支持。

[XXX CDN架构图]

过去几年，针对CDN的因特网流量和应用的研究达到了重要的里程碑[3-5]。例如，[3] 发现如今多数的跨域流量直接流经不同大型内容提供商，CDN和终端用户，其中30%的跨域流量产生自少数的几个内容提供商和CDN。一些研究也关注CDN的架构和性能，分析了例如CDN规模，服务器位置以及内容响应延时等特征[4-5]。其中，[4]重点研究Google CDN的内容响应延时，[5]对Akamai CDN的架构进行了深入的讨论。[17]则关注点在于YouTube服务中的异常检测问题。

- 点对点传输(P2P)：C/S架构中的服务器端是高负载的一段，其各方面性能必须要十分优越，否则很难处理峰值时刻的流量。随着用户数量的增长，服务器端的性能必须线性地增长。对于用户基数大的系统，一方面，为了处理峰值时刻的流量，服务器的成本将会十分昂贵；另一方面，在非峰值时刻，这些服务器却又造成了资源的浪费。为了应对这个困境，点对点传输通过充分利用客户端的资源来提供服务。在点对点网络中，每个客户端同时也是服务器，整个网络由所有客户端组成，客户端分别贡献出自己的网络带宽，磁盘空间等资源来分担部分的负载，共同支撑整个流媒体系统服务。

  P2P的几个关键特性使得其特别适合于实现内容管理和分发服务。首先，P2P网络通常不需要中心控制器，它实现了一个分布式算法用于网络管理，使得对等节点的失效不影响网络运作。其次，对等节点自愿加入网络被贡献相应的资源，除了网络访问之外没有任何额外的开销。最后，P2P网络原则上是可扩展的，因为每个对等节点在消耗网络能力的同时也做出了相应的贡献，因此网络总体上是可扩展的。不同的拓扑管理方式，形成了不同类型的P2P系统，这些拓扑用来路由相应的控制信息和内容。

  [XXX Pastry overlay]

  P2P系统已经广泛应用于在线视频点播服务之中。因为每个节点都提供服务，这是一种对大用户基数系统可扩展极强的架构。在[3-6]中，P2P的视频点播系统将对等节点随机的分配到不同的聚类之中，以提供视频检索服务；而在[7-9]中，对等节点形成了一个分布式哈希表；为了减少视频传

输/预取的延时，提出了将物理上相近的节点聚类在一起[7,10]，或者将拥有相似爱好的节点聚类在一起[9,12]。

2.2　流媒体个性化推荐

基于推荐的具体操作方法，视频推荐的相关工作可以被分为三类：基于内容的推荐，协同过滤以及混合过滤系统。基于内容的推荐算法将会推荐与用户曾经观看过的视频相似的视频，重点关注视频之间相似度的计算。例如，Mei等人提出上下文感知的视频推荐系统，VideoReach，该算法基于多模的内容相关性以及用户的反馈[Yang.2007XXX; Mei. 2011XXX]。它们使用两个视频文件之间的文本，视觉以及听觉信息来表示这个多模的关联，并使用注意力融合函数将它们关联起来。根据用户的反馈，融合过程的各个权重相应的调整。然而，这种分析视频内容的过程开销十分昂贵。协同过滤根据相似用户所喜欢的视频来推荐视频，因此，关键点就在于计算用户之间的相似度。例如，Baluja将用户的点击信息建模为user-video的二分图，并使用图算法“absorption”将偏好信息往整个网络中传播，据此完成个性化推荐。Davidson等人提出使用YouTube中的点击信息的关联规则来进行推荐[DavidSon.2010XXX]。park等人使用标签集合来作为用户的特征，并使用相似的观看模式来作出推荐[Park.2011XXX]。Zhao等人使用整合的观看历史来作出个性化推荐[Zhao.2012aXXX]。对于每个用户，该方法将对每个备选视频计算一个评分。评分包含两部分：好友对该视频的偏好程度，以及用户与好友之间的兴趣相似度。对视频偏好程度使用文本，视觉与流行程度相似度来计算；而用户之间的兴趣相似度使用各个用户的标记集合的相似度来估计。然而，稀疏性问题和冷启动问题仍然相纸的协同过滤的性能[Adomavicius and Tuzhilin 2005XXX]。总的来说，协同过滤关注用户之间的关系而忽略被推荐视频的特性，因此需要大量的用户和用户特征。另一方面，基于内容的方法更关注被推荐视频之间的关系以及视频的内容。混合过滤系统结合这两方面，例如，Zhao等人提出一个整合丰富信息的方式来进行视频推荐[Zhao.2011XXX]。它们将视频推荐视为一个排名问题，从多个信息源头中产生多个排名列表，最后使用整合算法产生最终的推荐列表。Cicekli等人基于图模型提出一个混合式的推荐系统[Cicekli.2011XXX]。首先使用adsorption产生推荐列表，最后使用基于内容的方法的结果替代推荐列表中的不相关视频。

主流的推荐算法采用协同过滤的思想。协同过滤的本质想法是：当我们要观察一个用户对一个视频的偏好情况的时候，找到与用户相似的其他用户，然后根据这些用户对该视频的评价来评估偏好情况；或者从另一个角度来看，可以找到该用户评价较高的其他视频，根据这些视频与该视频的相似度对偏好作

出估计。不论从哪个角度出发，最终都要落实到相似度的计算。在简单的模型中，可以使用每个用户或者视频的交互向量(有用户-视频交互的单元为1,否则为0)作为其特征向量，然后计算用户(视频)之间特征向量的余弦相似度,以此作为相似度，用于最后偏好值计算的加权系数。在上述计算中，我们假设每个用户(视频)的特征向量是一个交互向量，这种特征是从原始数据中得来的粗糙估计。在上述的例子中，使用的是用户交互数据作为特征向量，其中的每个单元格代表在某个视频上两个用户之间的0-1相似度。这种估计方法存在许多问题，一是每个单元格的估计数值不精确，使用一个实数域的值更加平滑；二是特征向量中包含了大量的单元格，每个单元格都代表一个视频的“投票”，这在实际运用中将导致存储大量的模型参数，而在大型网站中用户和视频的数量基数是以亿计的，存储这么大的特征向量是难以扩展的。为此，基于模型的推荐模型提出，用户的特征向量可以用一个维度较小的向量来表示，决定相似度的每个单元格不再是一个个实体交互数据，而是更加深层次与本质上的决定因素，这些交互数据不过是这些本质因素所反应出来的一种显示反馈。例如，在视频推荐的场合中，这些因素可能会是视频的各个吸引人的部分：动作成分，爱情成分，悬疑成分等待。我们称这些因素为“隐含因子”，因为它们并不在真实数据中体现出来，而是隐藏在真实数据之后控制着一切。因为在反馈数据中并未说明某个反馈是由于哪个因子产生，或者这些因子在产生该反馈的时候各占了多少权重，因此实际中我们只能得到每个实体(用户或者视频)的这些因子的分布，却不知道这些因子的真实类型，即我们知道一个因子对于决定A用户偏好十分重要，但却不知道它们的真实类型，只能用位置索引进行标识，即相同位置索引的因子代表同一类因子。上述讨论中，我们使用一系列有限的因子来表示用户的特征。而进一步，我们可以假设这些特征不局限于单个实体，而适用整个应用场景中的所有实体，例如对于视频的动作成分，用户也可以拥有其因子权重，代表有多喜欢某个类型的视频。这样，每类涉及到的实体都能拥有一个特征向量，用户视频之间的偏好值可以直接使用向量的内积来表示，最终的效果是每个用户和视频都被映射到一个特征空间之中，有几个因子就是几维空间。例如，预期中我们认为Peter不喜欢电影$alpha$，而更喜欢电影$beta$,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，这种隐含因子模型特别适合于存储开销和计算开销巨大的大数据集。给定一个实际系统的数据集，我们可以假设反馈数据产生的过程，通过该过程使用假设的模型去拟合反馈数据。但是另一方面，由于每个实体

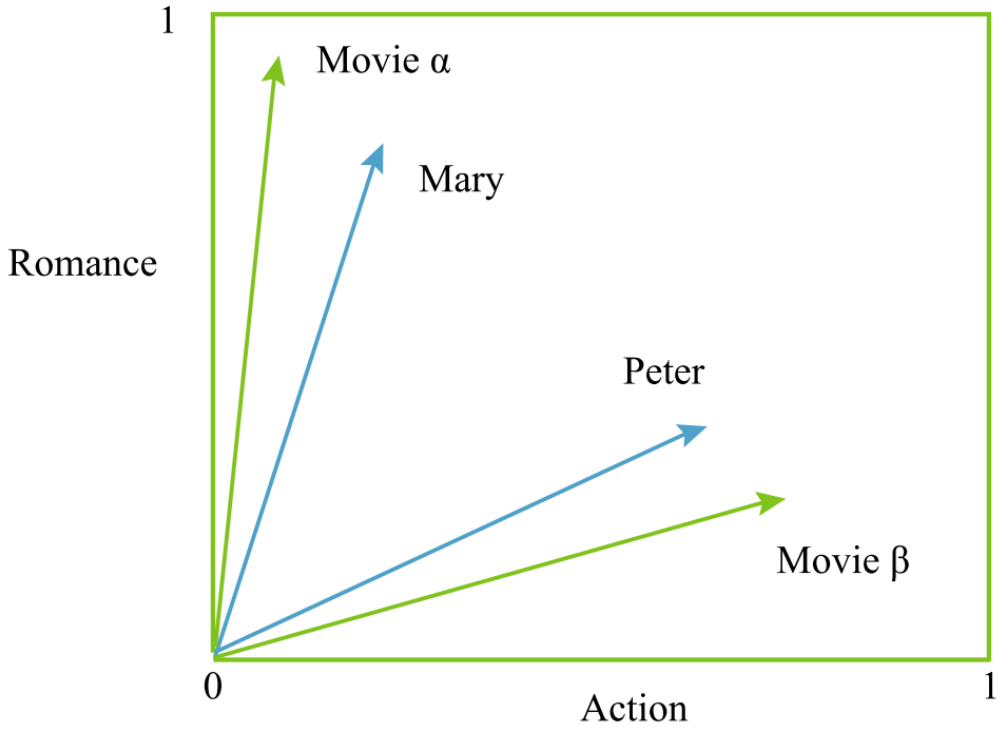

Figure 2.1: 依据隐含变量进行选择的过程

都拥有一组参数，因此模型的参数个数还是远远多于传统的模型。因此，在普遍使用该模型假设的推荐系统之中，存在着“稀疏性”和“冷启动”等麻烦的问题，会导致最终得到十分不精确的模型。典型的推荐系统中，实体的更新速度十分迅速，新的用户来去不断，每天都会有新的视频被上传。对于这些用户视频，历史记录稀少，由于协同过滤使用交互数据来从其他实体得到一个实体的特征向量，只有保证了一个实体大量的交互，才能得到一个准确的特征向量；同时，其他实体会参考这个不精确的实体得到各自的特征向量，进一步干扰了其他特征向量的训练。

为了应对这些问题，人们一方面致力于研究能够利用大量数据的有效模型，另一方面朝着利用不同领域数据的方向发展。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．

2.3　视频缓存

视频缓存包含服务端的缓存和客户端的缓存，在传统的互联网内容传输领域，大量的工作在内容传输网络(CDN)上进行研究[2][3]，同时有许多针对互联内容的缓存技术和缓存位置被研究[4-7]。互联网CDN，以及在其之上的缓存，并不能解决无线网络中视频的传输延迟和传输能力不足等问题。传统的缓存策略假设有大量的缓存空间，在移动场景中，这一假设也变得不再有效。已经有工作着手与研究在无线网络[8]和移动设备上[9]的网络内容缓存。然而，这些技术没有考虑传输和缓存视频所带来的特殊问题。在Ad-hoc网络中，缓存技术同样有被研究，如[10-11]。现有的大量工作集中于大视频的缓存工作，由于在用户观看行为总是遵循视频的播放时间，因此这类缓存任务很容易保证一定的缓存命中率，即使存在大量的VCR操作，估计出操作中频繁的视频片段，也能很好的进行预测。但是，在如今短视频盛行的网络环境下，考虑的用户观看的视频不应局限于一个视频，而是视频网站的任意一个网站，并且网站的视频数量远多于用户的设备缓存空间，那么传统的遵循时间顺序的方法就基本不可行。这时候，对缓存策略的设计必须更深入地考虑用户的行为模式和视频的传播模式，这也是本文结合缓存技术与推荐技术的一个根本原因。

第三章　基于多信号融合的分解推荐模型

3.1　引言

推荐系统是视频分享网站的重要一环，在内容泛滥的在线环境中，它能帮助用户快速检索到所需的内容。于普通用户而言，推荐系统帮助他们解除“信息过载”的困扰；于网站运营人员而言，推荐又是吸引用户的一种重要手段，由此为网站带来更多的收益。为了应付大量增长的视频和用户数量，隐含因子模型被用来建模用户的偏好与视频的属性，这种建模思路虽然能够精确描述用户的偏好属性，但是每个用户和视频都拥有一组参数，在典型的视频网站中存在“稀疏性”和“冷启动”等问题，这些问题的本质是训练数据不足以训练模型中的所有参数，造成模型不精确。

解决这些问题的方法显然是获取更多的数据，一个思路是获取与这些实体相关的其他应用场景下的数据，用于训练其在视频推荐场景下的特征。这需要保证两点，一是不同场景下的特征向量的相关性强，否则引入的数据只能被当做噪音处理；二是使用正确的模型来假设这些实体在其他场景下的行为，并且在建模的时候也当体现出与视频推荐场景有一定的相关性。例如，[**?**]假设用户在YouTube上的活动与其在Tweeter上的活动相关，使用Tweeter的信息训练推荐模型，这里额外使用的场景是社交网络。[**?**]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，这里购物场景和评价场景是涉及到的两个信息源。[**?**]将用户的GPS信息融合进基于位置服务的推荐系统，GPS属于日常生活场景，显然和位置服务属于不同的应用场景，但是利用它们之间的相关性依然可以获取有效信息。

在本章，我们采取另一个思路，即在一个应用场景之中获取不同类型的信号，使用这些信号对模型进行拟合。一方面同一个应用场景(都是视频网站应用),不同信号之间天然的具备一定的相关性；另一方面，这么做使得我们能够避免与其他应用场景打交道，却能得到更多的训练数据。具体而言，目前针对隐式的用户反馈，学术界普遍采用基于偏好偏序关系的学习方法获取模型参数，偏好的偏序关系是对推荐场景的良好类比，但是通常情况下用户的反馈数据极度匮乏，在仅存在正例和大量未知的反馈的情况下，良好的偏序关系极难获得。我们认为：这种偏序关系十分重要，通过偏序信号能够得到部分精确的特征向量，进一步利用视频的聚类效应，使得同一聚类中的视频尽量相近，能够使得部分不精确的模型向精确的模型逼近，进而得到更精确的模型。通过这种想法，

我们提出了基于聚类信号融合的推荐模型，其基本想法是融合偏序信号与聚类信号，从反馈数据中获取隐含特征。通过对真实数据的实验结果表明，聚类信号能够有效地提升推荐算法的效果。

在本章我们给出了我们基于聚类信息的推荐模型。首先介绍问题建模，提出了需要重点关注的几个问题并给出了简单的回答；之后给出算法的动机，在此基础上给出了算法框架的概览；然后分别介绍算法的两个组成部分：偏序信号模型和聚类信号模型；最后具体介绍模型的整合细节；最后介绍实验环节。

3.2 问题建模

推荐系统的基本目标是：为每个用户提供一系列的视频推荐列表。系统的输入包含“某个用户在哪些时间段观看了哪些视频”，使用这些信息，系统为每个未在历史记录中出现的⟨用户,视频⟩计算一个偏好值。对每个用户，系统将该用户未观看过的视频根据计算好的偏好值从高到低排序，之后推荐前几名的视频。这里面要解决几个问题：

- 如何表示每个用户和每个视频，最终如何计算偏好值？
- 如何处理原始数据以得到模型所需要的格式？
- 从模型假设到数据产生的过程是怎样的？

这些问题正是本章模型设计的关键部分，对这些问题的简单回答如下，具体详见后面介绍：

- 每个用户视频都使用一个向量来表示，我们称之为“隐含特征向量”，我们假设这些向量是控制反馈数据产生的所有参数，最后规定$\langle \hat{} r, \rangle$的偏好值通过相应的向量内积的来。
- 我们通过将原始数据处理成对应的“偏序信号”和“聚类信号”。偏序信号是$user, observed, unobserved$，代表“某个用户观看过某个视频却没观看过另一个视频”；聚类信号形式为$user, observed1, observed2, unobserved1$，代表“某个用户将observed1 和observed2聚类在一起，却没将observed1和unobserved1聚类在一起”。实际模型是通过拟合这些信号的来。
- 通过将原始数据处理得到相应的信号，我们将这些信号建模成“隐含特征向量”控制的随机事件(概率分布由隐含特征向量控制)，通过对系统中的所有偏序信号和聚类信号建模，得到这些信号的联合概率(使用了独立性

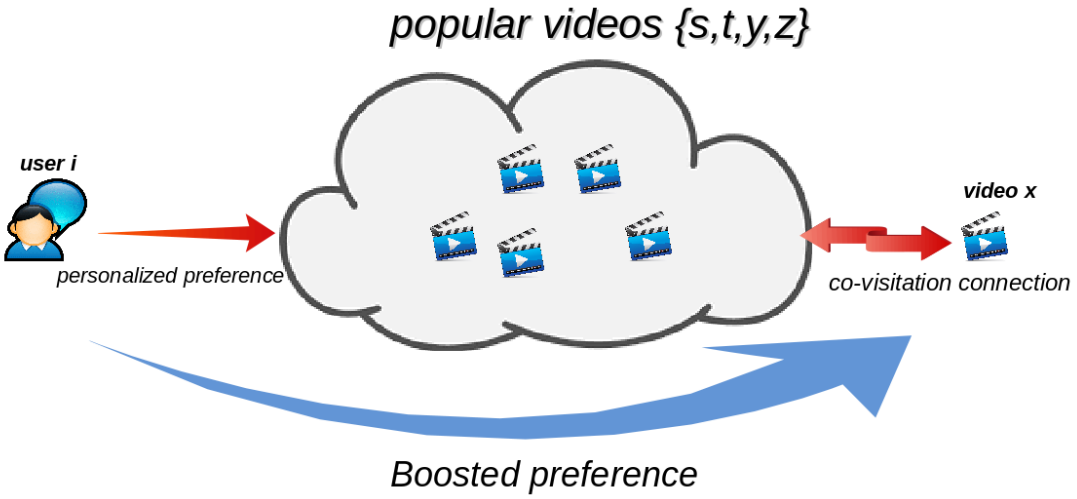

Figure 3.1: clustering boosted preference

假设)，得到后验概率最大值，得到描述这些信号的最好的“隐含特征向量”。因此，过程就是假设一个信号的产生过程，然后通过拟合真实数据得到最可能的那些模型参数，作为最后的模型参数。

3.3　算法简介

3.3.1　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3.1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推送到底部. 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.

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

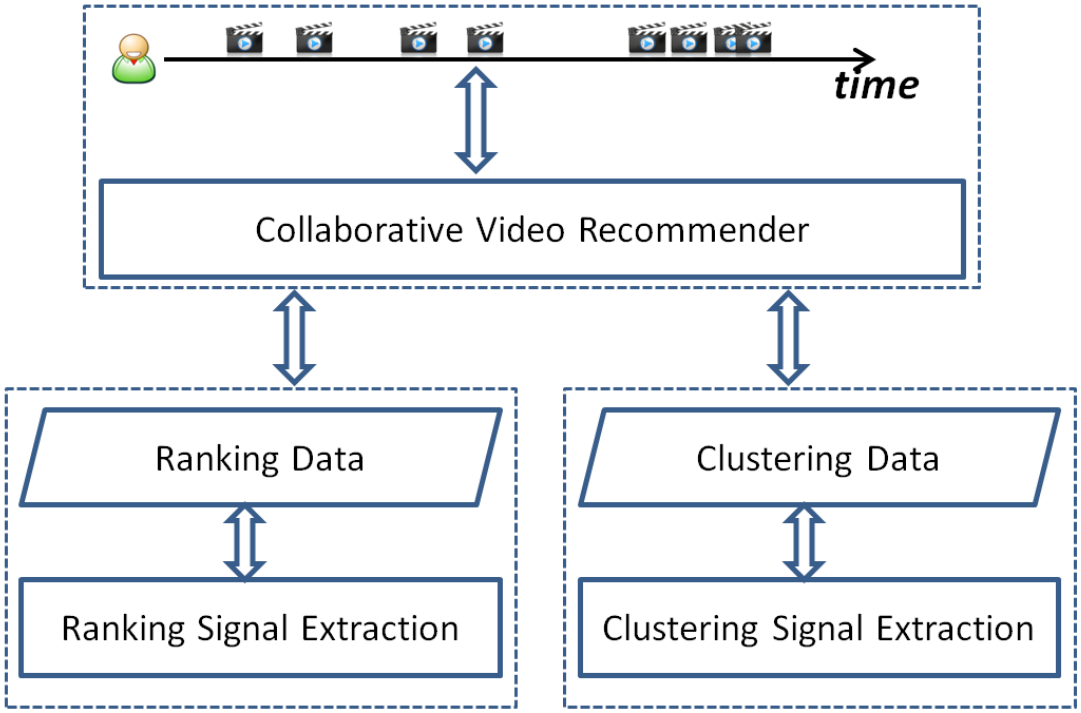

Figure 3.2: overview of model

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.

在整个推荐系统中，拥有的实体只有用户和视频，与它们之间的部分交互信息。推荐系统将在任何时间点，对每个用户提供一个视频的推荐列表，使得用户将来尽可能地在该列表选择要观看的视频。一次观看视频的会话包含被观看的一系列视频，传统的推荐系统只关注哪些视频被用户所观看，而哪些视频没有被观看，所有时间上的信息被简单地忽略。我们认为用户事实上无意识地将视频聚类在一起，这种聚类与用户的偏好高度相关，而时间戳信息告诉我们用户会如何将它们聚类。因此，我们提出一个“双信号融合”的推荐框架以充分利用额外的聚类信号，最终的损失函数包含偏序损失和聚类损失：

$$mix\_loss = ranking\_loss + clustering\_loss \tag{3.1}$$

模型之所以是“多信号”，在于原来的算法是将原始数据转化为一个个偏序对，即偏序信号，之后使用偏序对作为模型的反馈信息，这种偏序信号将作为我们的框架的信号源之一。在偏序信号之外，我们额外利用了原始数据中的视频聚类信息，即用户无意识地将视频在时间轴上进行的聚类，将聚类信息转化

为聚类信号，这个聚类信号将作为我们的算法的另外一个信息来源。最终的模型不但试图准确地预测原始数据中的偏好的偏序对，同时也尽可能的还原每个用户在时间轴上对视频的聚类，做法是使得对同一个聚类中的视频尽量给予相似的偏好值。

如图3.2所示，系统包含三个部分。对于每个正用户反馈，“ranking signal extraction” 模块将会采样一个负反馈。加上用户的标识，组成的三元组组成了最后的偏序信号。“clustering signal extraction” 根据时间戳寻找聚类，最终的聚类信号包含一系列的聚类对。将偏序信号和聚类信号最为输入，协同过滤推荐系统将会从偏序信号和聚类信号中获得训练模型所需的知识。第**??**和**??**分别介绍了偏序信号和聚类信号获取和建模的细节，第3.6节将会介绍整合模型的细节。

3.4　贝叶斯排序推荐

贝叶斯排序推荐(BPR)是针对隐式反馈数据进行优化的一种矩阵分解模型，其包括一种矩阵分解(隐含因子模型中的一种) 的建模方式和一种基于偏序对的优化方法。

3.4.1　形式化定义

假设U表示所有用户，而I表示所有视频,而所有反馈数据为$S \subset U \times I$。推荐的本质可以被看做是：给每个用户提供一个个性化的在所有视频上的全序$>_u \subset I$，其中$>_u$必须满足全序的性质：

$$\begin{aligned} \forall i,j \in I : i \neq j \Rightarrow i >_u i \vee j >_u i (\text{ø}S5) \\ \forall i,j \in I : i >_u j \wedge j >_u i \Rightarrow i = j (5) == \\ \forall i,j,k \in I : i >_u j \wedge j >_u k \Rightarrow i >_u k (D45) \end{aligned} \tag{3.2}$$

为了方便起见，做如下定义：

$$\begin{aligned} I_u^+ &:= \{i \in I : (u,i) \in S\} \\ U_i^+ &:= \{u \in U : (u,i) \in S\} \end{aligned} \tag{3.3}$$

3.4.2　基于偏序对的学习

由于通常推荐场景中只能得到正例反馈数据(用户点击，购买数据)，在所有未观测到的反馈中，混杂着未知的反馈与负例反馈。一种典型的做法是将所有反馈$(u,i) \in S$标为正，所有在$(U \times I \backslash S)$中的反馈视为负。因此，训练出来

的模型将会对观察到的反馈预测1，对其他的交互预测为0。这么做的问题是，对于在$(U \times I \backslash S)$中未来将要排序的单元来说，在训练过程中将会被视为负例。这意味着一个精确的模型将无法对这些单元进行排序，因为其预测偏好将都会为0。而这类机器学习方法能够预测的唯一原因是防止过拟合的技巧，例如归一化。

为此，BPR选择使用在视频上的偏序对作为优化的训练数据，这么做能够更好地表示推荐问题。BPR试图从S中恢复每个用户的部分$>_u$,如果一个视频i被用户u观看过，即$(u,i) \in S$，那么就假设用户对视频i的喜欢程度超过所有其他未看过的视频。如，在图3.3中，用户u_1看过视频i_2但是没看过i_1，因此就假设用户u对i_2的喜爱程度超过i_1。对于那些同时被一个用户观看过的视频，从中得不到任何偏好信息，同于那些同时未被一个用户观看过的视频而言，也是一样。形式化表示的训练数据如下：

$$D_S \coloneqq \{(u,i,j)|i \in I_u^+ \wedge j \in I \backslash I_u^+\} \tag{3.4}$$

这么做带来如下优势：

1. 训练数据中同时包含正例偏序对与负例偏序对。在为观看的两个视频之间的偏序值正是在将来需要排序的值。这意味着，从偏序对的角度来看，训练数据D_S和测试数据是不相交的两个集合

2. 由于使用观测到的偏序对$>_u$子集D_S进行训练，训练数据是从较为客观的排名上获得。

3.4.3　隐含因子模型

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．　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

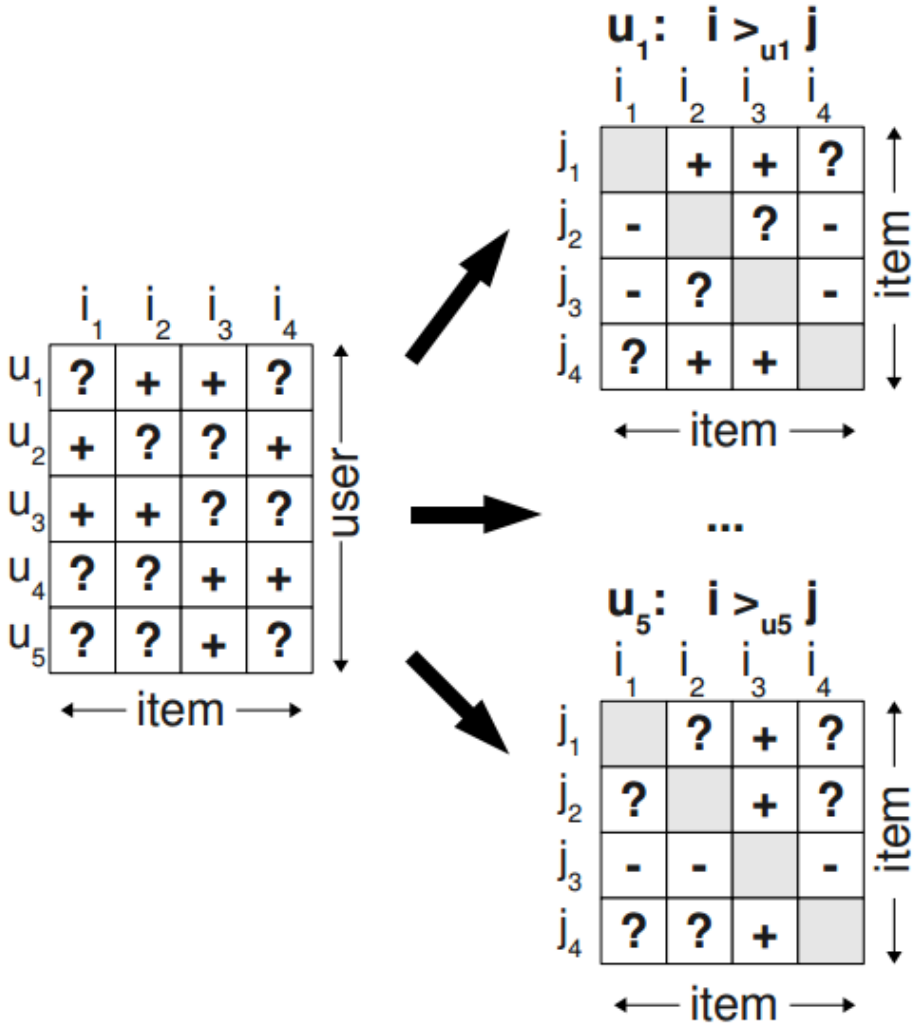

Figure 3.3: data processing of BPR

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：

$$h(j|i) = w_i + w_j + u_i^T v_j \tag{3.5}$$

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.

3.4.4　BPR 优化标准

我们的目标是寻找每个个性化的在所有视频$i \in I$上的正确排名。若Θ代表任意模型的参数向量(例如矩阵分解模型)，使用概率表示就是最大化下述的后验概率：

$$p(\Theta|>_u) \propto p(>_u|\Theta)p(\Theta) \tag{3.6}$$

这其中，$>_u$所希望得到的但是隐含的用户u的偏好结构。假设所有的用户的情况之间相互独立；同时假设针对每个用户，所有的偏序对(i,j)之间也相互独立。因此，上述的以用户为中心的似然函数$p(>_u|\Theta)$可以被重写为对每个用户和每个偏序对进行分解：

$$\begin{aligned}\prod_{u\in U} p(>_u|\Theta) = &\prod_{(u,i,j)\in U\times I\times I} p(i>_u j|\Theta)^{delta((u,i,j)\in D_S)} \\ &\cdot (1-p(i>_u j|\Theta))^{\delta((u,j,i)\notin D_S)}\end{aligned} \tag{3.7}$$

其中，δ是指示函数。根据全序性和非对称性，上述表述可以被简化为:

$$\prod_{u\in U} p(>_u|\Theta) = \prod_{(u,i,j)\in U\times I\times I} p(i>_u j|\Theta) \tag{3.8}$$

通过将每个偏序对表示为使用$sigma(x) = 1/(1+e^{-x})$函数转化的随机事件并且引入先验概率，最终的优化目标为

$$\begin{aligned}BPR-OPT &:= lnp(\Theta|>_u) \\ &= lnp(>_u|\Theta)p(\Theta) \\ &= ln\prod_{(u,i,j)\in D_S} \sigma(h_u(i|u)-h_u(j|u))p(\Theta) - \lambda_\Theta||\Theta||^2\end{aligned} \tag{3.9}$$

其中λ_Θ是模型特定的归一化参数。

为了优化该目标函数，可以使用梯度下降方法。然而，数据集中的扭曲的分布将会使得收敛速度极度下降。考虑一个流行的视频i，它在许多用户的记录中都存在，那么在每次梯度下降更新的时候，该视频i将会成为主导梯度的一个实体。为了使算法收敛，必须选择极小的学习速率。因此，更好的方法是使用随机梯度下降法进行优化。进一步的，更新每个交互反馈的顺序必须尽量随机，否则连续的更新将会涉及同一个视频或者用户，使得梯度值变小，降低收敛速度。由于数据集中的偏序对众多，实际更新时候，可以针对每个正例，采样一个负例，这么做能够使得收敛时间大幅度的减少。

3.5　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，我们希望也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．

3.5.1　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$f(\langle j,k\rangle|i)$,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：

$$\hat{f}(\langle j,k\rangle|i) = \theta_j^{\intercal}\Lambda_i\theta_k \tag{3.10}$$

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．

3.5.2　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3.4所示．

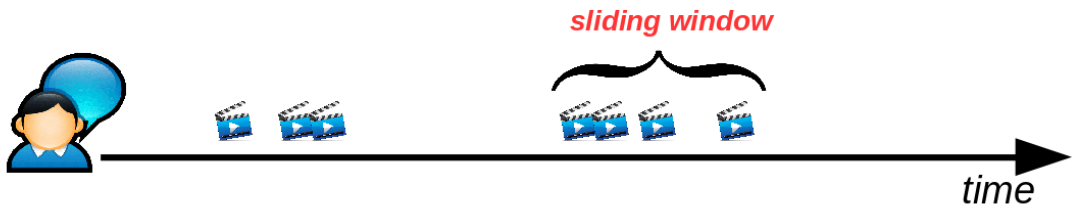

Figure 3.4: pairwise cluster extraction

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$\delta t = 2hour$．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．

3.5.3　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“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：

$$p(\langle j,k\rangle \succ \langle j,l\rangle) := \sigma(\hat{f}(\langle j,k\rangle|i) - \hat{f}(\langle j,l\rangle|i)), \tag{3.11}$$

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：

$$\underset{\theta,\Lambda}{\operatorname{argmax}} \prod_{(i,j,k,l)\in S_Y} p(\langle j,k\rangle \succ \langle j,l\rangle), \tag{3.12}$$

3.6　整合模型

3.6.1　信号的整合

图3.5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．

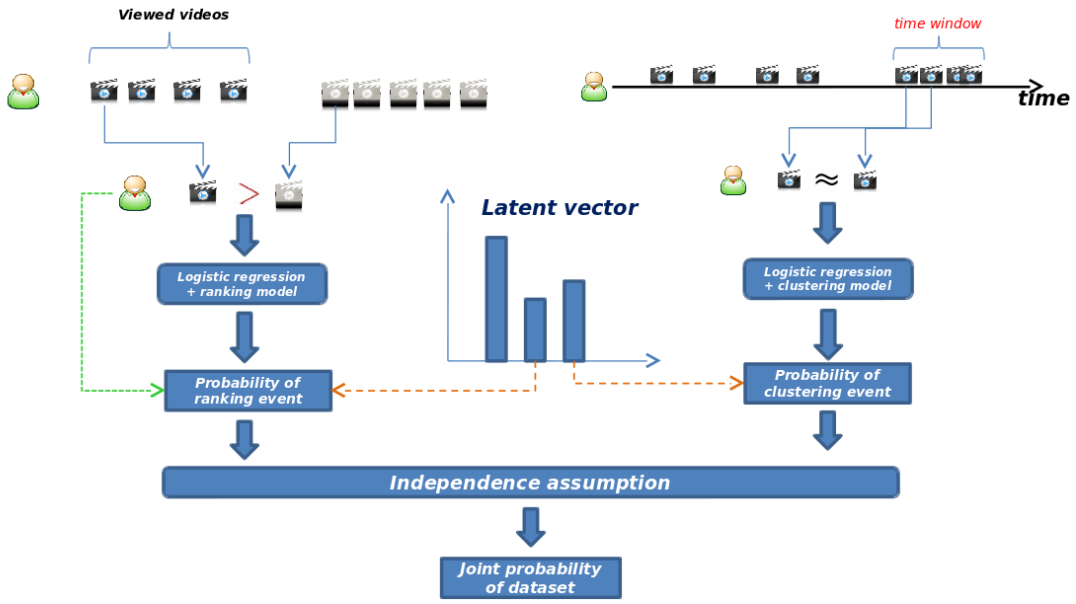

Figure 3.5: integration process

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

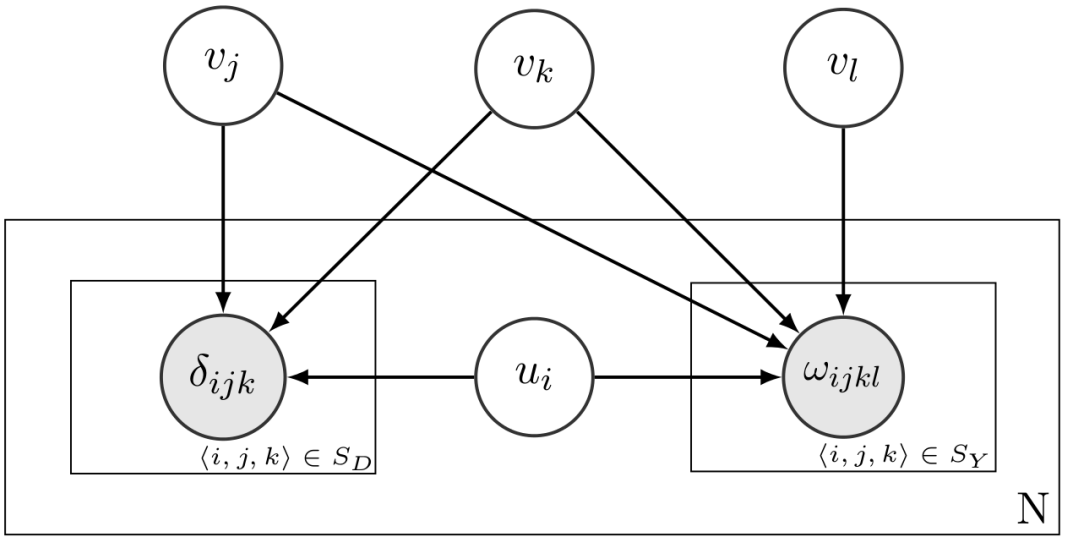

Figure 3.6: graphical model of ClusterRank

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．

3.6.2　模型描述

图3.6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．换句化说，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3.6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3.1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$\delta_{ijk} = 1 \sim Bernoulli(\rho_{ijk})$

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

其中ρ_{ijk}是“用户i对j的偏好大于对k的偏好”的概率:

$$\begin{aligned}\rho_{ijk} &= P(\langle i,j\rangle \succ \langle i,k\rangle | u_i, v_j, v_k) \\ &= P(\delta_{ijk} = 1 | u_i, v_j, v_k) \\ &= (1 + e^{-(\hat{h}(u_i,v_j) - \hat{h}(u_i,v_k))})^{-1}\end{aligned} \tag{3.13}$$

3. 对于每个聚类信号$\langle i,j,k,l\rangle \in S_Y$, 采样随机变量$\omega_{ijkl} = 1 \sim Bernoulli(\varphi_{ijkl})$
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：

$$\begin{aligned}\varphi_{ijkl} &= P(\langle i,j,k\rangle \succ \langle i,j,l\rangle | u_i, v_j, v_k, v_l) \\ &= P(\omega_{ijkl} = 1 | u_i, v_j, v_k, v_l) \\ &= (1 + e^{-(\hat{f}(u_i,v_j,v_k) - \hat{f}(u_i,v_j,v_l))})^{-1}\end{aligned} \tag{3.14}$$

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：

$$\begin{aligned}&P(\boldsymbol{U}, \boldsymbol{V} | \boldsymbol{\delta}, \boldsymbol{\omega}, \lambda_u, \lambda_v) \propto \\ &P(\boldsymbol{U}|\lambda_u) P(\boldsymbol{V}|\lambda_v) P(\boldsymbol{\delta}|\boldsymbol{U}, \boldsymbol{V}) P(\boldsymbol{\omega}|\boldsymbol{U}, \boldsymbol{V})\end{aligned} \tag{3.15}$$

使用最大化后验概率方法，我们得到如下目标函数：

$$\begin{aligned}\mathcal{L} = & \underbrace{-\sum_{\langle i,j,k\rangle \in S_D} \log(1+e^{\Delta\hat{h}})}_{\text{ranking data likelihood}} \\ & \underbrace{-\sum_{\langle i,j,k,l\rangle \in S_Y} \log(1+e^{\Delta\hat{f}})}_{\text{clustering data likelihood}} \\ & -\frac{\lambda_u}{2}\sum_{i=1}^{n} u_i^{\intercal}u_i - \frac{\lambda_v}{2}\sum_{i=1}^{m} v_i^{\intercal}v_i\end{aligned} \tag{3.16}$$

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

3.6.3　参数学习

公式 (3.16) 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：

$$\begin{aligned}\mathcal{L}^{(2)} = & \underbrace{-\sum_{\langle i,j,k\rangle \in S_D} \log(1+e^{\Delta\hat{h}})}_{\text{ranking data likelihood}} \\ & \underbrace{-\sum_{\langle i,j,k,l\rangle \in S_Y} \log(1+e^{\Delta\hat{f}})}_{\text{clustering data likelihood}} \\ & +\mathbb{1}_Y(\langle i,j,k\rangle)(-\frac{\alpha_u}{2}\sum_{i=1}^{n} u_i^\intercal u_i - \frac{\alpha_v}{2}\sum_{i=1}^{m} v_i^\intercal v_i) \\ & +\mathbb{1}_D(\langle i,j,k,l\rangle)(-\frac{\beta_u}{2}\sum_{i=1}^{n} u_i^\intercal u_i - \frac{\beta_v}{2}\sum_{i=1}^{m} v_i^\intercal v_i)\end{aligned} \tag{3.17}$$

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

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

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.

3.7 实验

3.7.1 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.

3.7.2 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：

精准度 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 精准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3.7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，

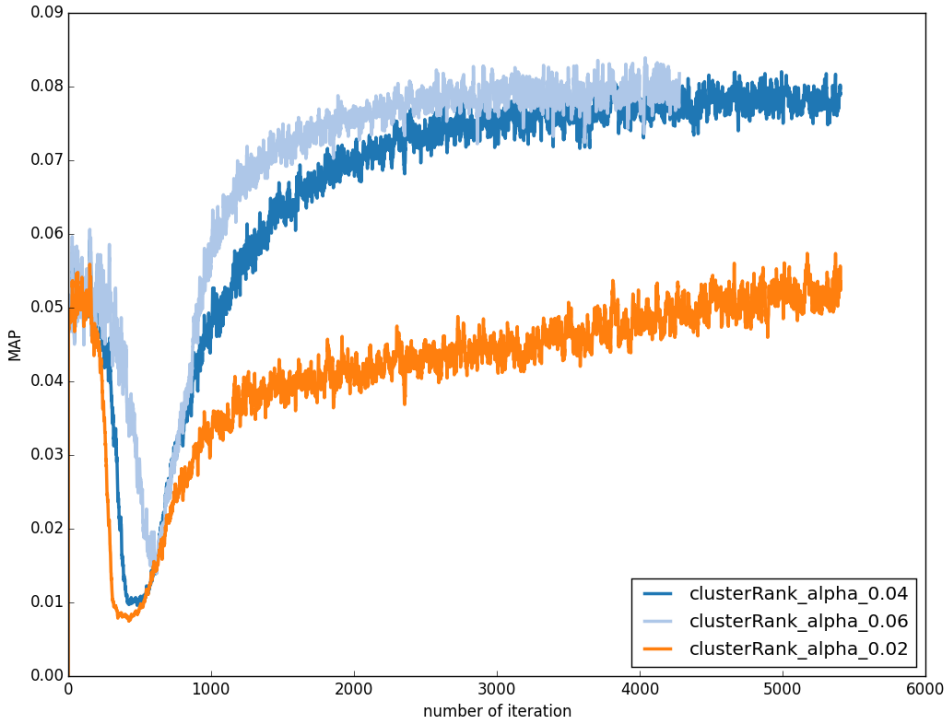

Figure 3.7: MAP for different $alpha$

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$\alpha = 0.06$.

3.7.3　结果分析

在图3.8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准度，召回率和平均精准度，两条曲线分别对应着BPR和我们算法的结果．对于精准度和召回率，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，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.

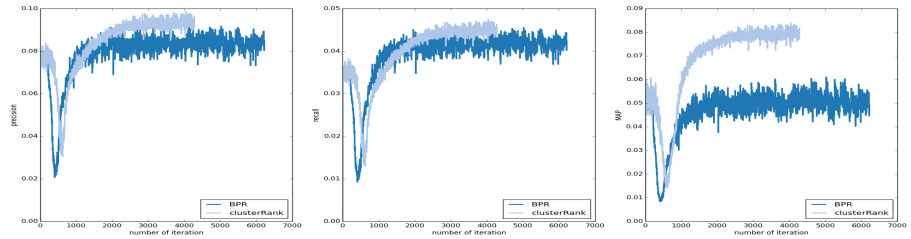

Figure 3.8: Performance for top-10 recommendation

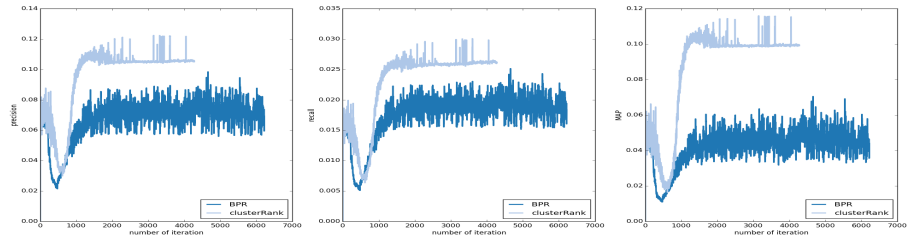

Figure 3.9: Performance for top-5 recommendation

如图3.9，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．

3.8　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　基于聚类推荐的视频缓存机制

基站传输方式存在固有的能力上限，在一个人口密集的区域，难以满足移动流媒体视频传输需求。因此设备之间传输被引入解决此类问题，移动自组网以设备之间传输为基础，建立一个移动网络，用于提供流媒体传输等服务。为了减少设备之间的信道争用，对视频进行缓存非常必要。为了保证足够的缓存命中率，需要精确评估用户对视频的偏好，本章基于视频推荐算法猜测用户对视频的偏好值，以此作为缓存的基础，并结合无线传输的广播特性，达到一次传输多次服务的效果。实验结果表明，使用基于推荐的缓存策略能够大幅减少视频传输的带宽占用。

本章给出了基于聚类推荐算法的视频缓存机制，首先描述使用推荐进行预缓存的动机；之后介绍我们缓存模块所依存的Ad-hoc流媒体分享系统；然后简单介绍推荐算法以及参数调节过程；然后描述推荐模块的系统结构；最后介绍实验结果。

4.1　动机描述

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．互联网视频的观看呈现着zipf分布，直白地讲，就是80%以上的视频请求集中于20%以下的视频内容之中，因此用户之间的偏好存在着很高的重叠，也是该缓存策略可行的一种保证。

4.2　基于智能手机的协同化移动流媒体系统

移动场景中，难以保证用户之间都在信号传输范围之内。为此，项目组基于MANET技术，设计并实现了“基于只能手机的协同化移动流媒体系统”。如图4.1，系统利用移动节点自身的通信能力，使得视频传输更加流畅，同时支持不断变化的网络拓扑。系统的架构如图4.2所示。

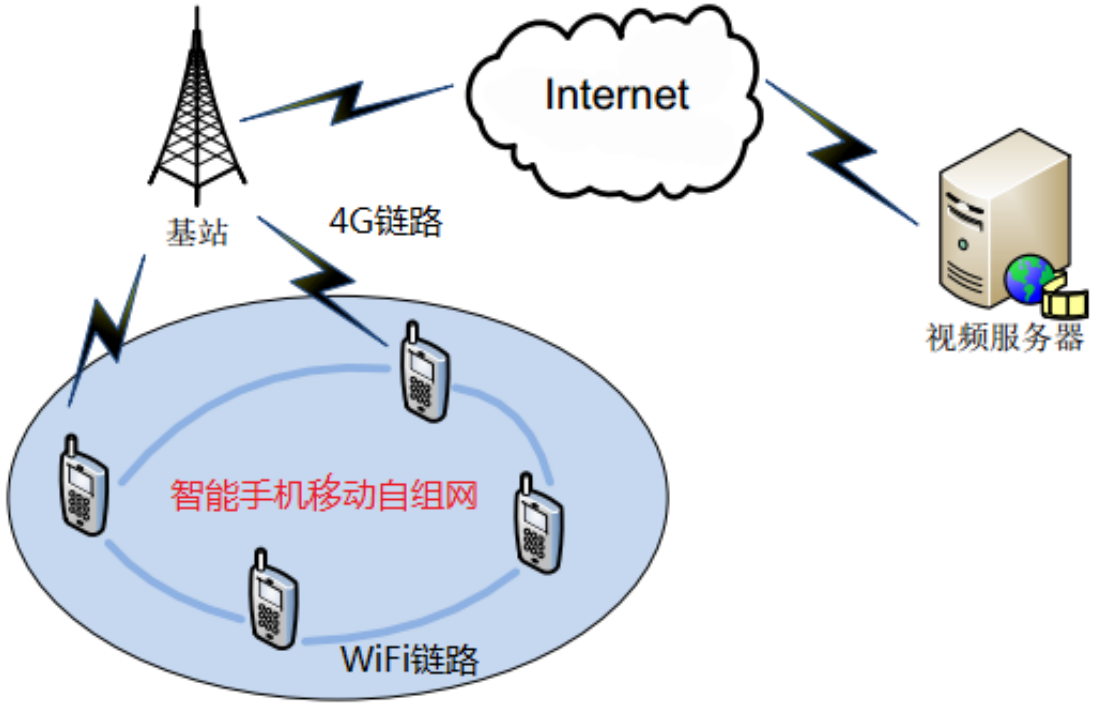

Figure 4.1: Ad-hoc网络系统架构图

在同一个区域的人将手机的网卡改为Ad-hoc模式，通过WiFi接口将手机进行互联。任务管理器负责对所有请求进行统一的管理。我们的缓存机制主要建

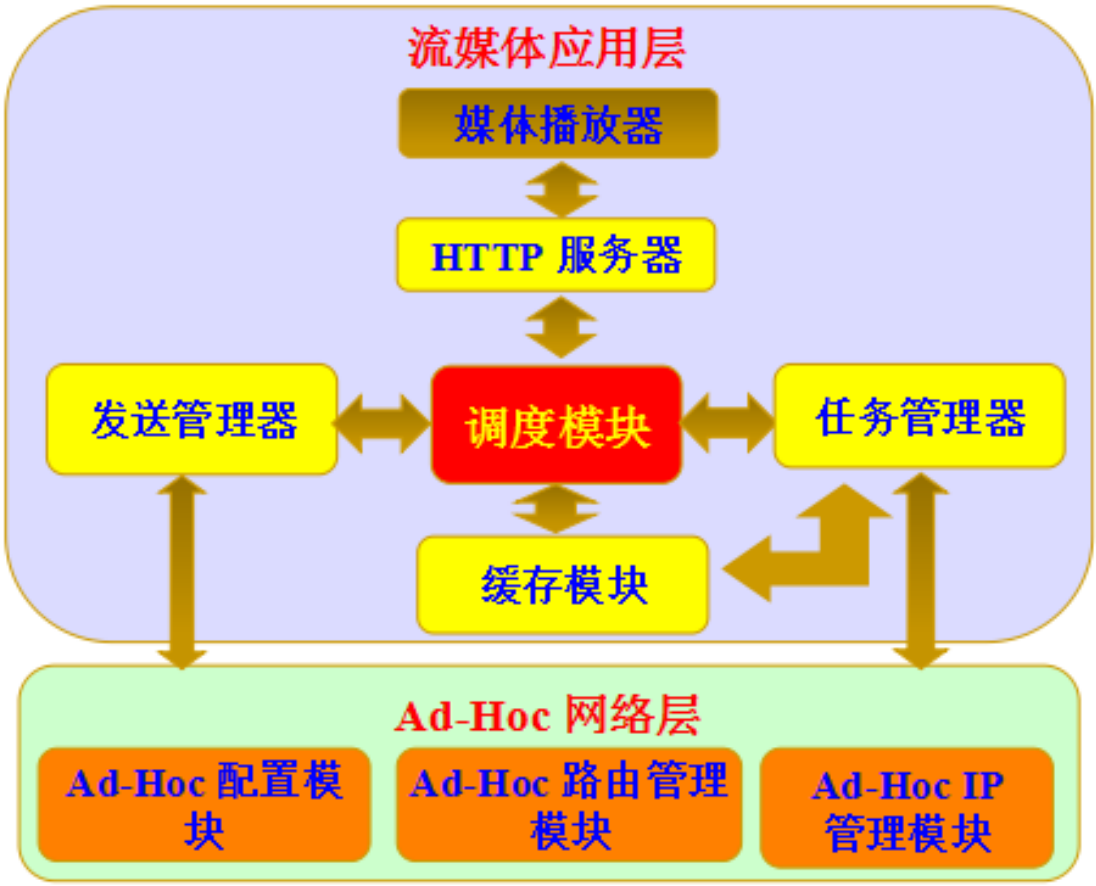

Figure 4.2: 系统模块架构图

立在缓存模块之中，缓存模块决定哪些视频需要缓存，并告知任务管理器，播放器需要显示缓存中已经存在的视频。缓存决策是由客户端的缓存模块和一个缓存推荐服务器共同完成。缓存模块定期与服务器联系，得到最新的模型参数，同时缓存模块将每一次观看行为记录，并将这些信息发送到服务器端作为训练数据。

网络层模块如图4.3所示，它是提供移动自组网下的网络层传输服务的模块，其底层的传输机制采用无线网络中的单跳UDP协议。IPManager负责维护各个节点的路由信息，处理Receiver中接收到的各种信息，同时在发送必要的通知消息。同时RouteManager负责维护各个节点的路由信息，包含路由消息格式和路由表格式，以及路由协议功能部分，路由协议采用的AODV(Ad-hoc On-demand Distance Vector)路由协议。该模块同样与Sender和Receiver 模块交互，并使用同步队列进行同步。Node模块包含RouteManager与IPManager，负责协调各模块工作，同时接收来自应用层的请求。应用层通过Node提供的接口传输与接收数据，底层对应用层透明。

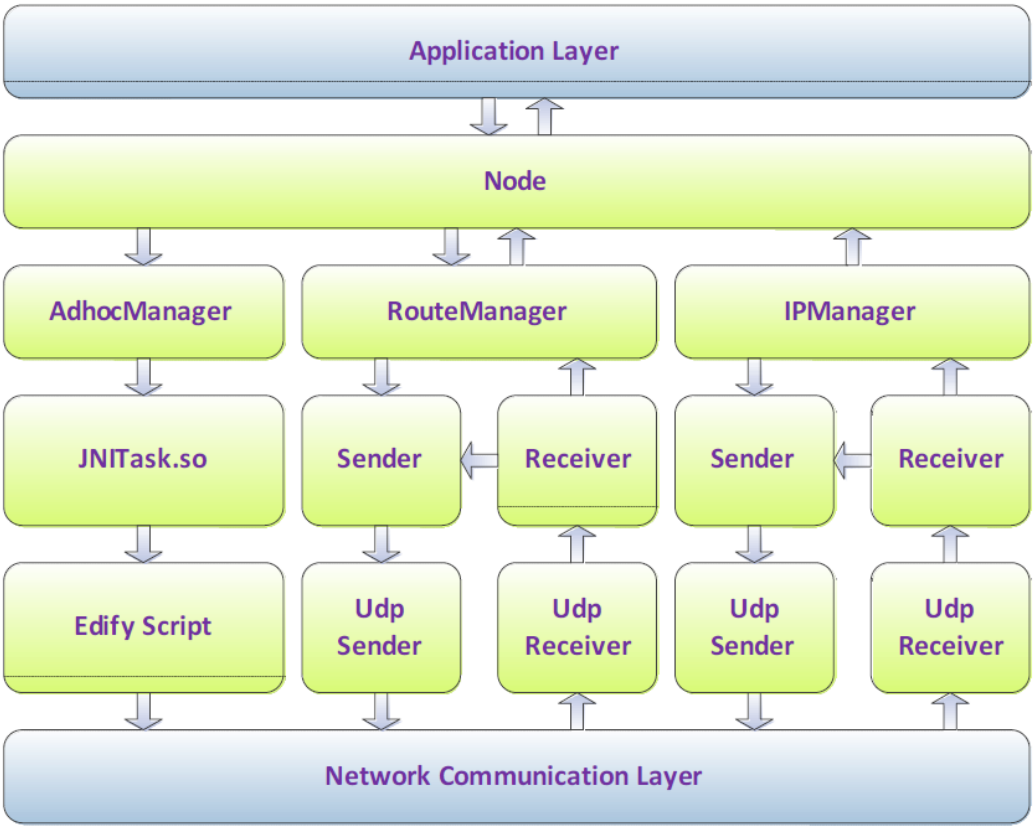

Figure 4.3: 网络层架构图

移动设备上保留两类缓存空间，一类是视频播放过程中的播放缓冲，另一类是预缓存所需要的视频缓存。在开始观看之后，用户首先在推荐列表中选择所观看的视频。若选择观看了推荐视频，那么所有的视频内容将从预缓存空间中获取。否则用户查询本地播放缓存，如果还不存在，则通过移动自组网获取数据，若依然不再，最后通过蜂窝网络获取视频数据。其中预缓存空间的内容的更新与获取都由推荐缓存模块来实现。

4.3　用于缓存的推荐算法

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：

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部视频的场合。

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。

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4.4所示，在使用加权策略之后，精准率有了明显的提高。

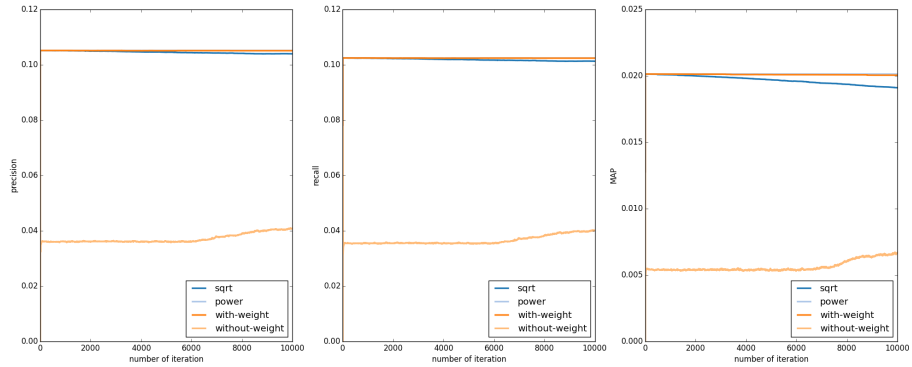

Figure 4.4: 加权vs 不加权效果对比

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4.5给出了结果对比，与预期一样，聚类

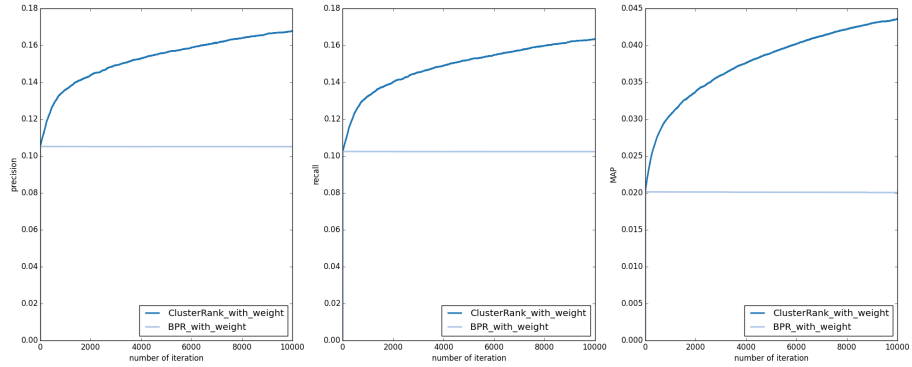

Figure 4.5: 加权情况下的效果对比

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。

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，客户端和服务器. 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)；服务器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.

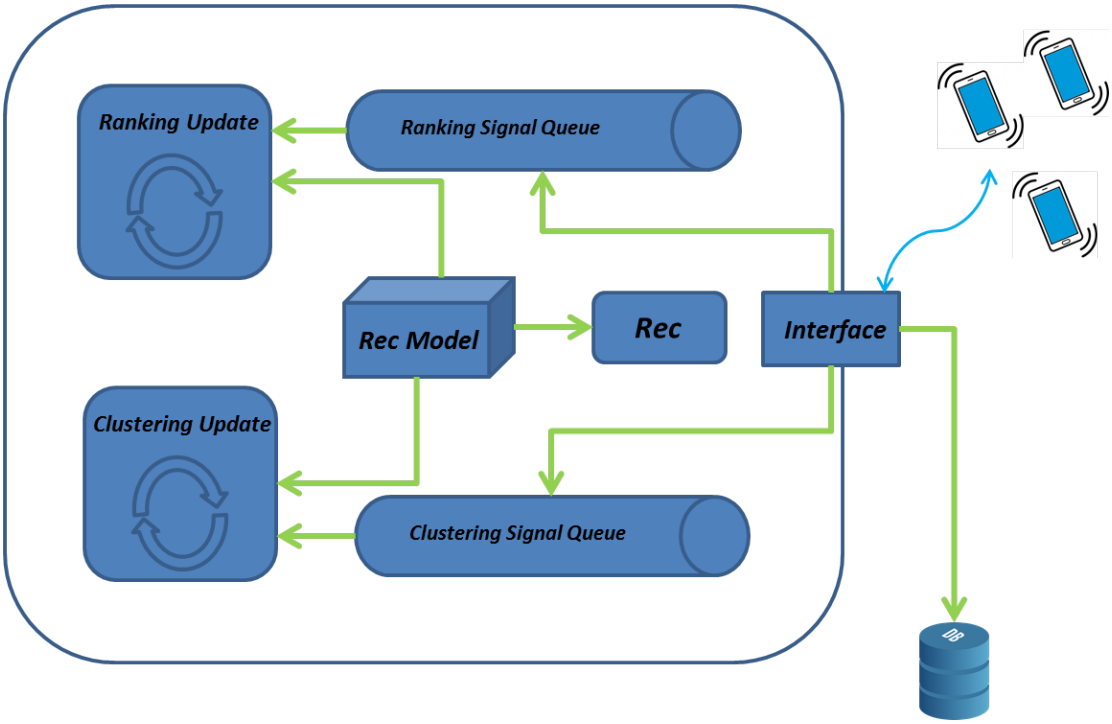

Figure 4.6: 推荐预取系统框架图

图4.6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的任意的．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，新的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

$$communication\_save = \frac{(hit\_num - distinct\_videos)}{10 \times user\_num} \tag{4.1}$$

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

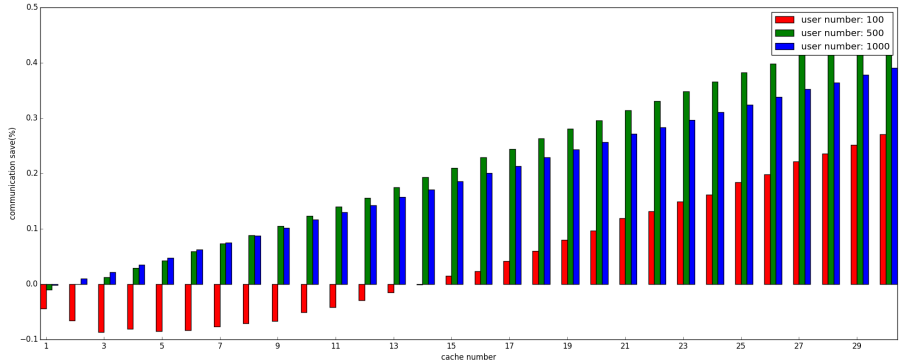

Figure 4.7: 传输节省随缓存数量变化

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4.7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，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

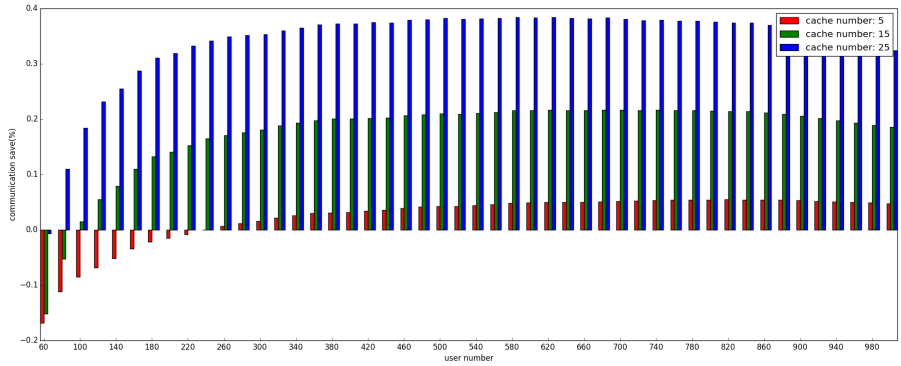

Figure 4.8: 传输节省随用户数量改变

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4.8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．

4.5.6　结果：不同方法比较

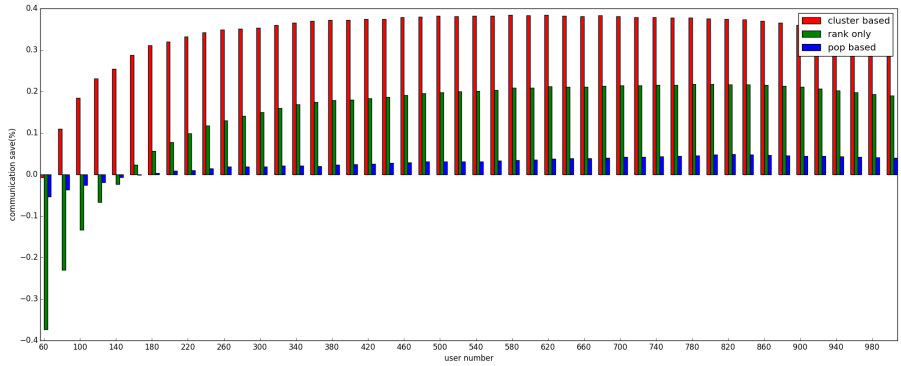

Figure 4.9: 不同方法的比较

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