PaperPass检测报告简明打印版

比对结果（相似度）：
总　体：12 % *（总体相似度是指本地库、互联网的综合比对结果）*
本地库：12 % *（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）*
互联网：0 % *（互联网相似度是指论文与互联网资源的比对结果）*

编　号：VIP90A977DC076A9BB5D
标　题：推荐系统
作　者：李朝阳
长　度：23231 字符(不计空格)
句子数：823句
时　间：2015-4-15 19:18:01
比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源
查真伪：http://www.paperpass.com/check.aspx

句子相似度分布图：

本地库相似资源列表（学术期刊、学位论文、会议论文）：

1. **相似度：1 % 篇名:《协同过滤算法在移动电子商务推荐系统中的应用研究》**
   来源：学位论文 厦门大学 2013 作者: 杨波
2. **相似度：1 % 篇名:《基于协同过滤模型与隐语义模型的推荐系统研究与实现》**
   来源：学位论文 湖南大学 2013 作者: 鲁权
3. **相似度：1 % 篇名:《基于用户标注动机与遗忘曲线的个性化推荐研究》**
   来源：学位论文 重庆大学 2013 作者: 陈曦
4. **相似度：1 % 篇名:《推荐系统中协同过滤算法的研究及应用》**
   来源：学位论文 浙江大学 2012 作者: 弭真真
5. **相似度：1 % 篇名:《群落标签推荐系统体系结构及关键问题研究》**
   来源：学位论文 南开大学 2012 作者: 董振华
6. **相似度：1 % 篇名:《基于协同过滤的个性化推荐算法的研究与应用》**
   来源：学位论文 江苏大学 2013 作者: 包增辉
7. **相似度：1 % 篇名:《基于Web挖掘的个性化推荐算法研究》**
   来源：学术期刊 《计算机与数字工程》 2014年4期 作者: 杨艳霞 于海平 陈燕
8. **相似度：1 % 篇名:《基于社会网络的主动信息推送算法研究》**
   来源：学位论文 杭州师范大学 2012 作者: 余天豪

9. 相似度：1％ 篇名:《基于时间因子的动态推荐算法研究》
来源：学位论文 中南民族大学 2013 作者: 许斐
10. 相似度：1％ 篇名:《混合协同过滤个性化推荐算法研究》
来源：学术期刊 《计算机光盘软件与应用》 2014年4期 作者: 黄琼 冯军焕
11. 相似度：1％ 篇名:《个性化文档推荐系统的设计与实现》
来源：学位论文 华中科技大学 2013 作者: 杨玲
12. 相似度：1％ 篇名:《基于信任传播模型的协同过滤推荐算法研究》
来源：学位论文 中山大学 2010 作者: 陈晓城
13. 相似度：1％ 篇名:《一种改进的基于物质扩散理论的Item-based协同过滤算法》
来源：学术期刊 《数字通信》 2013年2期 作者: 刘群 陈阳 易佳
14. 相似度：1％ 篇名:《基于本体的影视个性化推荐算法研究》
来源：学位论文 武汉理工大学 2013 作者: 魏欢
15. 相似度：1％ 篇名:《基于预测置信度的协同过滤稀疏性问题研究》
来源：学位论文 中国科学技术大学 2013 作者: 程小林
16. 相似度：1％ 篇名:《基于级联二部图的动态推荐算法》
来源：学术期刊 《计算机工程与设计》 2013年12期 作者: 蒋宗礼 陆晨
17. 相似度：1％ 篇名:《基于主题模型的英语写作批阅系统个性化推荐模块设计与实现》
来源：学术期刊 《科技和产业》 2013年6期 作者: 葛昊 叶艳 包西林 吴敏
18. 相似度：1％ 篇名:《协同过滤推荐研究综述》
来源：学术期刊 《微型机与应用》 2013年6期 作者: 张瑶 陈维斌 傅顺开

互联网相似资源列表：

没有找到与互联网相似度高的资源！

全文简明报告：

{60％： 推荐系统为解决信息过载的提供了一种重要的方式;} {42％： 它通过分析用户行为日志预测用户的喜好，将来的行为以及兴趣等并为用户推荐物品。} 随着互联网的不断发展，用户的参与度越来越高，而早期推荐系统主要基于内容或者基于人口统计的推荐，将而这些不能在再满足当下用户实时数据无法产生实时推荐的需求。 而且在以往的推荐系统中并没有关于整体推荐系统设计实践进行描述的文章。 因此本文基于公开用户行为数据集对推荐系统进行优化，在原有算法的基础上添加了时间变量，并验证算法的实效性。 本文的主要工作内容如下:

文中为推荐系统建立模型，并优化了对应的算法从而实现了用户对未评分物品评分的计算。 大体即通过相关的方法提取用户特征，计算用户物品的预测评分。 同时在考虑时间因素在预测中的影响，判断用户的喜好程度不仅与物品的属性有关， {49％：而且与用户所处的上下文也有关系，用户在不同的时间，不同的地点对物品喜好程度也是有区别的，} 而物品在不同的环境中的热门程度也是不同的。

相对于海量的物品，用户购买的或者用户感兴趣的物品数量仅仅占总量的很少一部分。 {47％：要从所有的物品中找到用户感兴趣的物品集合，然后在对物品集合中未评分的物品进行评分。} {46％：本文分析了基于用户隐形数据上的Top-N推荐算法，主要就是问了得到一个用户感兴趣的集合；}

{41％：另一部分就是关于推荐系统的设计与实现，其中包括推荐引擎，日志系统，VIEW系统。} 推荐引

擎主要是算法的组织和算法的可扩展方面的设计，一个优秀的推荐系统只靠一个或者固定的几个推荐算法是不可能实现的，所以要考虑引擎的可扩展，还有就是单一业务方面的考虑；日志系统主要记录用户的行为日志（显性和隐形），还有一些用户规则如：30岁的互联网女性喜欢购买零食等；以及一些推荐结果；而VIEW系统主要分为两方面：用户行为日志的收集，通过调查问卷，物品打分以及商品浏览次数统计等方式；再有就是推荐列表的展示，远程接口提供第三方展示，插件展示等提供多种展示方式。

关键词：{45%：推荐系统，日志系统，VIEW系统，推荐引擎}

Abstract

Recommended system is an important way to solve the information overload. It’s recommended for users log items by analyzing user behavior to predict the user's preferences, future behavior and interests, etc. With the development of web2.0, and increasing user involvement, But, the earlier recommendation system based primarily on content or recommendations based on demographics will not be able to meet the current user real- time data cannot be generated in real-time recommendation needs. In the past, and no information on the recommended system overall recommendation system design practices described in the article. Therefore, this paper based on publicly available data set of user behavior is recommended to optimize the system, including real time user dynamic recommendation.

The main contents of this paper are as follows:

Users of the goods were scored by knowing item’s scores. By extracting features that users related methods to calculate user's predicted score goods. Also considering the time factor in the prediction of the determined level of user preference items is not only related to the attribute, but also with the context in which the user has a relationship, the user at different times, different places like article is also a difference, while the popularity of goods in a different context is different.

Relative to the mass of the goods, the number of items purchased by the user or the user is interested only accounts for a small part of the total. All goods in the collection to find goods of interest from the user, and then did not score for the goods in the collection of goods scored. This paper analyzes the Top-N recommendation algorithm based on user data on stealth, the main interest is to ask a set to get a user;

Relative to the mass of the goods, the number of items purchased by the user or the user is interested only accounts for a small part of the total. All goods in the collection to find goods of interest from the user, and then did not score for the goods in the collection of goods scored. This paper analyzes the Top-N recommendation algorithm based on user data on stealth, the main interest is to ask a set to get a user;

Key Words： Recommender System, Logging System, View System, Recommender Engine

目录

图目录

表目录

绪论

课题背景与意义

随着web2.0 的快速发展， 无论是用户产生的数据， 还是物品的数量已经从匮乏到了过载的阶段， 那么如何从海量的物品中找到真正用户需要的物品;

早期的互联网， 人们通过收藏夹或其他方式记住网址，来查找所需要的信息， 随着时间的增长， 收藏夹里的内容也越来越多， 用户在收藏夹中也慢慢的将网址进行了分类。 同时，互联网中也将信息进行了分类， 这里做的最好的就是雅虎公司了。 但是随着信息的不断增长，分类目录不能随着信息增长的速度而增长， 分类目录只能囊括互联网很少一部分的信息。 {42%： 而且维护成本也不断的增加， 于是产生了新一代的信息检索技术：} 搜索引擎。

{96%： 人们提供信息的关键字，通过搜索引擎检索自己所需信息；} 这里面有一个难题， 就是用户如何准确的表达自己的意思， 或者用户意图的关键字是什么，搜索引擎就是根据用户的关键词在索引系统中查询与之匹配的信息的； 在某些场景中用户是无法描述自己的需求的， 在这种情况下，用户如何获取自己所需要的信息； {46%： 这就是我们本文要提出的一种方案， 推荐系统。} {45%： 它通过分析用户的行为来预测用户需要什么样的信息， 然后将信息反馈给用户。} 用户在根据自己的喜好来评价这些推荐信息，从而能更加准确的为用户提供推荐。 这样在信息获取方面， 推荐系统和搜索引擎形成了一种互补的形式。 {43%： 搜索引擎是用户主动获取信息， 而推荐系统是用户被动的接受用户所需要的信息。}

在信息的海洋中，需找你自己都不能明确的信息， 何其难？ 在网站中上架的商品何其多，怎样才能将你需要的商品展现在你的面前，这给电子商务平台带来了具体的挑战。 如何将与用户相关的商品推送到用户的面前，实现信息的长尾理论。 {59%： 这种将用户和商品联系起来的系统即为推荐系统。}

{54%： 个性化推荐系统，解决了信息生产者和信息消费者之间的通信障碍。} {45%： 从消费者角度来看，系统将消费者想要的信息推送给了消费者；} 而消息的生产者想将信息发送给需要的用户， 同样系统也满足了生

产者的需求。 个性化推荐的实质就是解决长尾理论，而双11是国内最有名的电商促销活动。 在双11大促会场上应用个性化与否，成交金额可以相差50%，这对一个对电商平台来说，绝不能忽视的增量。 而个性化技术在双11中主要应用的地方有搜索个性化、推荐产品优化。 在双11中如何里利用个性化推荐技术，提高消费者购物体验，增加长尾商家和商品的销售，提高平台的收益，成为一个非常关键的点。 {48%：通过下面的内容，我们一起来探究这些问题。}

研究内容

{56%：推荐系统的主要目的是，为用户发送该用户喜欢的信息；} 这就涉及两部分： {44%：1）如何找到与用户实时行为相关的物品；} 2）如何展现相关推荐并与用户交互产生实时数据； 本位主要探究以上两个方面。

{47%：协同过滤算法是第一代推荐系统也是应用最为成熟的推荐系统，目前有两类主要的协同过滤算法：} {92%：基于用户的协同过滤算法和基于物品的协同过滤算法。} {47%：基于用户的系统过滤算法就好比，用户A有一批志同道合的朋友，他们对物品 I都有兴趣并做了评分，} {45%：而推荐系统就根据 A的朋友的评分进行计算判断是否为 A推荐物品 I；} 其核心思想是： {69%：利用用户行为数据计算用户间的相似性；} {58%：得到与用户相似性较高的邻居-用户好友群，然后利用这些用户对其物品的评价来预测目标用户的喜好程度，然后做出推荐；} 基于项目的协同过滤算法是： 用户A对物品集合有一个相似的评价，当需要对某物品I进行预测评估的时候，可根据对相似物品的集合的评价来对物品I进行推荐。 这种推荐算法适用于电影，音乐等方面，这些类型的物品都是难以结构化的类型。 还有就是与用户无关。

{60%：基于内容的推荐算法在推荐界的应用最为广泛。} 它的工作流程大体为： {50%：为每一个物品建立物品属性列表；} {53%：为每一个用户也建立一个用户兴趣列表；} {56%：然后计算用户兴趣属性列表与物品属性列表之间的相似度；} {62%：相似度高的证明用户对物品感兴趣，那么就将该物品推荐给用户。} {43%：这种算法的优点在于为用户建立了兴趣模型能准确的为用户提供推荐列表。} {41%：同时也存在一些瑕疵， 1. 关于算法的冷启动问题；} 2. 对物品和用户建模的问题； {53%：这个模型好坏直接影响推荐结果的质量。}

公开用户数据集合

本文主要依据以下这些数据集合进行推荐算法的实验验证。 这些数据集合中有部分数据集合经过进一步的处理， 对实验的结果有一定的影响。

亚马逊（z.cn）数据集

从亚马孙网站上抓取的数据， 其中包括548552种不同的产品（书籍，音乐CD，DVD和VHS录像带）的信息。每条记录包括题目， 销售排行， 相似产品列表， 产品的分类以及产品的评价。

{43%：图 1 1数据集所含各种类型物品数目}

MovieLens电影评分数据集

{44%：MovieLens 数据集描述了电影推荐服务5星评级和免费文本标签的活动。} 它包含了100023收视率和整个8570电影2488标签应用。 {44%：这些数据是从1996年4月2日到2015年3月30日间创造的706用户。}

CiteULike论文书签数据集

CiteULike是一个着名的论文书签网站。 它允许用户保存和共享引用学术文章。 并且让上传论文的作者给论文打上标签。 CiteULike提供数据集合中包含作者，论文，论文标签还有时间四个字段，而这些刚好满足我们实验对数据属性的要求。

论文概要结构描述

{51%：本文除第一章之外可分为五部分：}

{42%：第二章为预测评分设计了一个模型；} {63%：通过预测算法预测给定用户对给定物品的评分；} {45%：再有就是计算用户物品评价预测值的方法介绍。}

第三章描述过去的Top-N推荐，并在此基础上添加了用户的短期兴趣，而后建立了新的模型以及改进了Top-N推荐。

第四章推荐系统的整体需求分析， 将整个系统分为三个子系统， 然后分别对三个子系统进行了需求分析，功能概括。

{55%：第五章推荐系统的具体设计和技术实现；} 推荐引擎的整体架构； VIEW系统的框架设计，收集用户行为日志的接口设计，为用户推荐展示的方式以及远程调用等接口的设计与实现。

{49%：第六章论文结束语以及对推荐系统的展望}

{49%：评分预测中用户物品特征的模型}

用户物品特征描述

{48%：特征， 在某一领域内能够代表一个实体的描述。} 一种可以利用领域知识生成和提取特征， 特征的提取一般通过和销售或者行业的专家进行商讨， 听取他们对特定业务的意见，并提取相关的特性， 当然这个特性越多越好； 例如， 从程序员购买的电脑的配置来看，程序员对电脑的配置要求都比较高，因此我们可以将此电脑的配置作为特性来衡量用户兴趣。 {44%：当然也可以通过统计信息分析的方式提取用户特征；} {45%：本章主要是对简单用户商品交互特性进行建模；}

推荐系统依赖用户特征， 没有这个基础， 推荐系统无法为用户推荐物品，它是推荐系统必不可少的一部分。一般通过让用户为物品评分来表达用户的兴趣，同时用户评分行为被称为显性反馈行为，指用户的行为明确指定倾向于指用户喜欢/不喜欢该物品。

{54%：表2 1具有代表性的显性反馈数据}

网站显性反馈

视频网站对视频的评分

{52%：新闻类网站对新闻的评价等}

{54%：购物网站对商品的评分以及评价}

{45%：音乐网站对音乐/歌手/专辑的评分}

{89%：显性反馈行为，用户明确表示对物品喜好的行为。} 网站收集用户的显性信息的方式也各不相同，如使用表情图标表达用户的喜好程度等，但是大多数网上在收集用户信息上使用5分评分体系。 天猫商城就是将5分评价体系用到极致的网站，对商品的描述相符，服务态度，发货速度，物流速度，快递员的服务态度这些都使用了5分评价体系。 这些使用5分可以很客观的描述用户对物品的喜好程度，但是有些网站就不大推崇5分体系，比如豆瓣的FM频道，他们使用 “喜欢这首歌/不再播放” 来描述用户对这首音乐的喜好程度，对于音乐来说，用户只能说是喜欢或不喜欢，如果在细致划分喜欢的程度是多少，这也就太难为消费者了，并不是每个用户都是音乐专家。 收集用户行为信息的方式多种多样没有哪种是最好的，只有适合需求的才是最好的。

为用户推荐物品必须要有依据其中给用户评分是最为常用的一种方式，但是用户如何给那些没有评分的物品评分呢？ 给用户建立特征模型即可，我们10年前喜欢的物品和现在我们喜欢的物品差异巨大，或者两者之间根本不存在联系，比如我们之前喜欢动画片，现在呢？ 可定动画片不是你所喜欢的； 这就是时间的魅力，如果在推荐系统中不引入时间而进行推荐那么这些推荐的结果实在不敢恭维， 此类的时间上下文因素在互联网中有很多，而如何利用这些时间因素提高系统预测的精度， 设计符合用户物品特征变化的动态推荐系统，是近年来推荐领域研究的热门问题。

本节的主要内容如下： 我们使用矩阵分解模型来计算用户特征关系，而本节主要介绍的就是静态的矩阵分解模型以及用奇异值分解算法来计算用户对未评分物品的评分。

{52%：推荐系统的静态用户特征矩阵模型}

{40%：静态用户兴趣模型，即与时间因素无关的用户兴趣特征模型，矩阵分解模型也可以认为是对用户-物品二维模型的数据不全问题。} {46%：详细说明一下，用户对物品的评分仅仅占居了用户，物品的很小的比例。} 无论是单个用户对物品的评价总数占物品总数的比例，还是多个用户对一个物品的评价中多个用户与用户总数的比例； 这些都是很小的比例； {61%：因此组成的矩阵可以说是一个稀疏矩阵；} {100%：我们现在的任务就是讲此稀疏矩阵所有坐标都给补全了。}

问题数学定义

{48%：定义D={(u，i，r)}为用户u对物品i的评分的集合，设有N名用户，M件物品，我们组成一个RN*M的二维稀疏矩阵；} 在图2-1中我们看到一个简单的用户物品评分矩阵，一行代表一个用户对所有物品的评分，空格代表没有评分，也是需要我们计算的。 比如，u1对i1，i4有评分，但是对于i2，i3，i5都没有评分。 {50%：下面我们通过奇异值分解算法来计算用户评分。}

图 2 1用户-物品评分矩阵

奇异值分解算法（SVD）

该算法的主要计算步骤是： 首先为每个缺失位置初始化一个初始值，比如一个用户对所有用户评分之和的平均值：

R(u，i)=（∑？ r_ui）/(|N(i)|) (2.1)

{46％：其中N(i)代表对物品进行评价的个数；}

{49％：或者也可以通过多个用户对一个物品的评分之和的平均值：}

R(u，i)=(∑？ r_ui)/(|N(u)|) (2.2)

{56％：其中N(u)代表为物品i进行评分的用户的个数。} {47％：将评分补全之后的矩阵记为R？}

然后SVD将R？ 分解为三个矩阵相乘：

R？ =U^T*S*V (2.3)

其中U∈R^(K× N)，V∈R^(K× M)，S∈R^(k×k)是三个分级或的矩阵。 S为对角矩阵，其对角线上是R？矩阵的特征值。

然后在在R？ 中挑出x个最大值，将其组成对角阵Sx，同理得到Ux，Vx。 最终得：

(R？ =U_x*S_x)？ *V_x (2.4)

其中R？ 即为评分矩阵R的最终补全矩阵，R？ (u，i)即为用户对物品的预测评分。

本章小结

本章主要是计算特定用户给明确的物品打分，通俗的说就是预测用户给某一物品的评分。 用户和物品之间没有直接的关系，但是用户有过评分记录，同时用户的好友也有评分记录，那么计算用户与物品（用户没有评分）之间的评分。

Top-N推荐用户兴趣预测的模型

Top-N推荐简介

为一个用户产生一个推荐列表大体需要两步，第一步是得到一个和用户相关的物品列表，在真正的生产中系统中包含很多件物品， 但是真正和用户相关的物品很少，我们仅仅只要基于与用户相关的物品即可； {46％：第二步是在得到用户将会评分的列表之后，预测用户会给该物品多少评分。} 第一步则是本章要解决的内容，该问题也可以说是Top-N推荐问题； 关于Top-N推荐的研究有很多。 {48％：其中协同过滤是很多生产系统中的主要算法[，]。}

{47％：用户的行为数据根据是否是用户主动返回数据可将数据分为显性反馈数据，隐形反馈数据；} 显性反馈数据是用户通过评分体系等明显的方式主动返回的数据，比如，对物品的喜欢/不喜欢，评分，评价等，但是

返回的数量比较少； 而对于隐形反馈数据，比如用户浏览商品的记录， 网站不可能在用户浏览物品之后让用户标记自己浏览过来物品， 而这种数据就是隐形数据。 这种数据不能明确用户是否喜欢物品， 但是用户关注物品了， 我们就可以通过这种关系为用户推荐与物品相关的推荐。 还有一种数据， 就是在一天中某一段时间的流量， 我们以分钟为单位来统计， 网站每分钟进入网站的流量，同时也可以观察流浪进来的渠道，是联通的多，还是电信的多， 是早上的流量高还是晚上的流量多； {52％： 从这些数据我们可以进一步为用户提供更加优质的服务。} 还有智能家居中面包机和咖啡机的开启时间，室内光线强弱的调整。 这些我们都可以通过用户反馈的隐形数据来给用户推荐执行。 {43％： 期间用户根本不用明确的为推荐系统提供反馈数据。} 这就是隐形反馈数据的优势， 当然文献[ ]中对这两种数据的优劣势做了详细的分析，虽然没有显性反馈数据那么明显的优势， 但是仔细思考它具有的某些特性也是显性数据所不具备的。 在特定的场景下， 隐形反馈数据也是能够反映用户的特性的。 {41％： 本节主要描述利用隐形反馈数据来过滤和用户有关系的物品列表， 即Top-N推荐。}

问题数学定 义和概括

传统Top-N推荐

一般的数据集Data = {(user， item)}，其中数据集中的每个二元组(user， item)代表用户 user对物品 item产生过行为。 Top-N推荐的目的： {68％： 为用户推荐用户最感兴趣的物品；} {57％： Top-N的方法就是为用户建立用户兴趣模型。}

基于时间的Top-N推荐

{46％： 用户的兴趣是随着时间变化的，并且用户的历史行为也随之增加。} {43％： 这时用户的兴趣随着发生一定的偏移，长期兴趣不变， 但是短期兴趣很有可能随着时间的变化而转移。} 为了适应用户兴趣的变化，推荐结果也随之发生变化，这变化最主要是要考虑时间因素， 本章要解决的问题是在给定时间内，为用户提供用户感兴趣的物品。

{74％： 推荐方法-基于图模型的协同过滤算法}

{49％： 利用二分图为用户行为建模。} {51％： 模型中U代表用户的集合， I代表物品的集合， 如果Ux对Iy产生了动作行为， 那么E代表两者之间的行为，而行为的强度定义为边的权重w(e)。} 如图3-1所示， 图中定义了U1， U2， U3 和 I1， I2， I3， I4之间的关系。 {51％： 从图3-1中我们可以看到每对顶点代表一个用户和物品之间的关系；} 比如 U1 和 I1， I4之间有行为； U2对 I1， I3， I4有行为； U3对I2， I3有行为。

图 3 1用户-物品关系图

{47％： Top-N推荐的主要任务就是找到和用户有行为的物品集；} 反映到二分关系模型上就转化为了用户节点到物品节点是否存在路径， 有几条路径， 路径的度为多少； 比如，在图2中用户 U1到 I2，之间并没有直接关系，但是有 U1→ I4→ U3→ I2这条路径到达 I2， 也就是说 U1和 I2，之间有一条长短为3的关系。 计算图中顶点之间距离的算法有很多，Fouss在文[ ]中总结了很多， 我们介绍一种路径融合算法[ ]

路径融合算法

{56％： 算法的基本思想是判断两个节点之间的相似度，其标准是：}

在用户物品关系集合里有多条边可以连通两个节点；

{53%：再者就是两顶点之间存在相对较短的路径；}

{42%：尽量避免在顶点之间路径中出现出度很大（热门物品）的节点。}

从以上三个标准中可以得到计算节点间相似度的过程，第一步我们先找到所有能够连通两个顶点之间的路径，并计算每条路径的权重；{41%：第二步比较权重和节点相连需要经过的边数找出最小的；} 第三步若出现节点出度很大的节点，并且有其他路径选择，则舍弃出度大的那条路径。

以图3-2为例。{42%：为A用户推荐物品，从图可知A对物品i1，i3产生过了行为，那么推荐物品只能在i2，i4之间产生；} 然后我们从图观察看 A对 i2，A对 i3是否有路径，结果是 A到 i2有（A，i1，B，i4，D，i2）和（A，i1，C，i4，D，i2）两条路径；而A到i4有（A，i1，B，i4）和（A，i1，C，i4）两条路径；{45%：根据第一条标准我们找到了到两个节点的所有路径。} 然后计算所有路径的边数。{44%：A到i2，之间的两个路径的边数都是5条；} A到i4的两条路径的边数是3条。{45%：根据第二条标准，用户对物品i4的兴趣相比物品i2的兴趣跟大一些；} 然后在比较A到i4之间所经过的节点的出度的大小，B的出度为3，C的出度为2；最有的路径即为（A，i1，C，i4）；为了更加准确的计算用户对物品的相似度，我们可以规定当节点的出度不大于某值的时候，将所有的到节点i4的路径相加求平均，来精确用户对武平的相似度。

图 3 2 路径融合算法

通过上面例子的详细描述，我们可以更加精确的描述路径融合算法了，从而用数学公式来计算用户对物品的相似度也就是用户是否对物品感兴趣。{47%：首先计算用户到物品的最短路径，之后再计算最短路径的权重；} {50%：最后将所有最短路径的权重相加求出用户的物品的相似度。}

下面就是路径融合算法的数学表达：

计算两点之间的路径权重，其中两点之间边的个数和经过的顶点的个数对权重的大小起着决定性的作用，数学定义： 设P为两点之间的路径，σ(V_n)? [0，1]为顶点V的权重，ω(v，v')? [0，1]为顶点v和v'之间边的权重；那么P的权重定义为：

γ(P)=σ(Vn)∏_(i=1)^(n-1)? (σ(Vi)? ω(V_i+V_(i+1)))/(|out(Vi)|ρ) (3.1)

其中σ(V)? [0，1]是顶点V的出度。从公式是可得出当n越大，也就是边的数量越多，那么P的权重也就越小。

我们在求两点之间的权重的时候是遍历求取图中点与点之间的权值，其实变相的也求得了用户和物品间的相似度了。以下是数学定义：{51%：p(V_1，V_n)为V1和Vn之间路径的集合，求V1 和 Vn 之间的相似度。} 那么V1和Vn 的相似度为：

d(V_1，V_n)= ∑_(P∈P(V_1，V_n))? 〖γ(P)〗 (3.2)

本节的模型中并没有加入时间因素，在下文中我们将用时间和行为数据建立模型，{49%：再次使用路径融合算法计算用户物品相似度，即用户对物品的兴趣程度。}

隐性反馈数据的动态变化

用户的长期兴趣和短期兴趣

人类的兴趣集合中有很多兴趣点，他并不是固定不变的，长期兴趣只是相对于短期兴趣来说，关注长期兴趣的时间比短期兴趣更长一点。比如一位驴友他的长期兴趣就是游山玩水，但是，他也会关注实时的政治消息。在上述的场景中，旅游属于用户的长期兴趣，而政治新闻只是属于用户受其他因素影响而产生的兴趣，属于短期。{42％：因此，人的长期兴趣是历史积累而逐渐发展而来的，相对时间较长；}而短期兴趣的产生受到各方面的影响，你的朋友对品茶特别的擅长，你通过和他的接触慢慢产生了对品茶的兴趣，这是一个短期的过程。{45％：这就使得我们在推荐的时候通过时间将用户的短期兴趣和长期兴趣都加入用户特征中去了。}{42％：我们可以根据长期兴趣短期兴趣时间等建立一个新的模型，进而改进算法适应该模型。}

在二分图中引入长期兴趣和短期兴趣，我们只需要引入时间节点即可，之前是用户对物品产生兴趣，那么用户和物品之间有边相连，并有相应的权重。加入时间后就可以这样表达用户对物品在T时段有兴趣，T大于6个月那么即为用户的长期兴趣，T小于3个月即为短期兴趣，当然时间是可以调整的。{41％：这样根据时间的长短控制用户的兴趣，从而计算用户与物品之间的兴趣程度了。}

因为引入了新的变量，原有的模型名字也就不能再表达新模型的意图了，我们这里通过一个新的名字来表示该二分图即时间段-图模型(Time based Graph Model(TGM))；TGM是一个二分图，定义为G(U，S，I，E，w)，其中U，I，E与上文路径融合算法中的含义相同，S代表新的变量即时间段；这也是该模型的一个亮点。{59％：下面我们通过一个简单的例子来说明该模型的定义。}

在图3-3中A，B代表用户；而A：1，A：2，B：1，B：2这种代表用户在1或者2时间段；用户可以用A，A：1，A：{53％：2到物品的路径代表用户对物品的长期兴趣，用A:}{57％：1代表用户对物品的短期兴趣；}从图中可以看出，A：1对i1感兴趣；A：2对i2感兴趣；A对物品i1，i2，都感兴趣；如果计算A：{50％：1对物品i3是否感兴趣的时候，A:}1到达i3的结果必然是A在1时段的兴趣点是i3；{61％：从而实现根据用户短期兴趣进行推荐。}而从A出发到达i3则是代表与A的物品集合相似的物品，也就是代表A的长期兴趣是i3。也就是说我们可以通过控制A和A：{45％：1两个节点的权重值从而控制用户的长期和短期兴趣在推荐结果的占比。}TGM中一共有三种节点：A，A：1以及i1；它们的权重可以为：

$$\sigma(v)=\begin{cases}1; & v\in I@\beta; \\ & v\in U@1-\beta; \\ & v\in S)\end{cases} \qquad (3.3)$$

其中β？{42％：[0，1]，调整β的大小来调节推荐结果中短期兴趣所占比重。}

图 3 3 时间段-图模型

基于TGM的路径融合推荐算法

我们建立了时间段-图模型，并解释了它的合理性，现在就该改进我们原有的路径融合算法，从而适应性的模型。

基于TGM的路径融合算法主要思想是：计算用户物品相似度在加入了用户短期兴趣因素之后，例如下面这个简单的例子，

{50%：在图3-3中，计算用户A在2时段和物品i3的相似度。} {48%：如果考虑用户A到物品i3的长期兴趣，那么首先就要计算d(A，} {42%：i3)的相似度，然后再计算A在2时段的相似度，最后将长期兴趣和短期兴趣相加即可得到s(A2，} i3)的相似度。

计算过程如下：

计算A和i3之间的相似度:

d(A，i3)= γ(A，i1，B，i3)+γ(A，i2，B，i3)+γ(A，i2，B2，i3) (3.4)

其中γ(p)见上节公式，下面计算A在2时段对i3的相似度：

d(A2，i3)=γ(A2，i2，B2，i3)+γ(A2，i2，B，i3) (3.5)

{43%：最后计算A在短期兴趣也就是2（月）时间内容对物品i3的兴趣程度：}

s(A2，i3)=d(A，i3)+d(A2，i3) (3.6)

{51%：根据实例我们可以推导出计算用户物品相似度的通用公式如下：}

s(u，i)=d(u，i)+d(u_t，i) (3.7)

其中d(u，i)为相似度。

本章小结

{53%：本章首先对用户物品关系建模，并通过路径融合算法计算用户对物品的相似度；} 但最主要的是在原有的基础上引入了用户的短期兴趣，通过时间属性来表达从而建立了一个新的模型即时间段-图模型，{40%：从而使得用户的短期兴趣在推荐结果中也可以体现，同时用户的推荐结果随着时间的变化而变化；} 当然，有了新的模型同时也需要一个新的路径融合算法来计算用户的相似度，在文中我们详细的介绍了新的路径融合计算方法。

推荐系统需求分析

推荐系统概述

推荐系统具体的可以分为三部分，分别是推荐引擎（推荐子系统），日志系统，VIEW系统；系统通过VIEW系统收集用户的行为日志，将用户行为日志做进一步的处理，并持久化到日志系统中；而后由推荐引擎从日志系统中抓取数据，计算用户兴趣，并为用户输出包含N件物品的推荐列表；再由VIEW系统展示推荐列表；这是整个推荐系统的大体架构。

本节的主要部分在于：

推荐引擎的需求分析； 该子系统的任务就是从由日志系统产生的数据中拿到以清洗的数据， 然后通过用户特征分析， 生成用户的特征向量。 不过如果是非行为特征，就不需要使用行为提取和分析模块了。 {44％： 该模块的输出本来就是用户特征向量；} 然后通过矩阵转化将特征向量转化为推荐列表。 最后对初始的推荐列表进行优化，去除不符合逻辑和规定的，再将结果进行一定的排序形成最终推荐列表。 {47％： 主要对上述过程进行了详细的功能划分。}

VIEW系统的需求分析； {42％： VIEW系统是推荐系统中最终结果的呈现， 同时也是优化推荐结果的窗口；} 这部分主要工作内容是划分整个系统的各个功能模块，并详细的描述了那些主体功能。

日志系统的概述； 描述了日志系统的职责，并将日志系统根据系统所需职责将其划分为几个部分。

推荐引擎概述与需求分析

系统目标

推荐子系统的结构我们采用模块化的设计方式， 一个模块代表一个推荐引擎， 一个推荐引擎仅仅负 责一个任务，或者一种特征类型。 {43％： 最后的推荐结果是将各个推荐引擎的推荐结果按照一定的权重合并或者选取一个最为符合的推荐结果。} 这样做有以下几种优点：

推荐引擎可以随意的增加和删除，这种设计符合软件开发中的“ 开-闭原则 ”， 最终的推荐结果是几个推荐引擎共同作用产生的，每个引擎在其中所占的权重是可以配置的。

{44％： 这里的推荐引擎并不一定代表推荐算法， 也可以是一种推荐规则或者推荐策略。} 比如程序员喜欢配置较高的笔记本，这属于一种推荐策略。 或者用户喜欢根据人口统计信息进行推荐的推荐结果，或者喜欢好友推荐，或者喜欢一种推荐算法的推荐结果，我们可以通过用户的喜好反馈为用户配置不同的推荐权重。 从而提高用户推荐精准度。

系统功能性需求描述

生成用户特征向量

{46％： 我们一般从用户的注册信息中抓取用户的特征。} {47％： 注册信息中包含了大多数关于用户的人口统计学特征。} {41％： 这种特征我们可以直接拿过来直接生成特征向量。} {94％： 特征向量就是由特征和特征的权重组成；} {52％： 还有一种特征是从用户的行为数据中计算得来的。} {42％： 在利用行为数据计算用户特征向量的时候需要考虑以下因素：}

用户行为的种类 用户在网站中的操作有好多种， 浏览， 点击， 提交， 购买， 对比等等； 用户对物品有各种各样的行为， 我们应该选取哪种行为作为特征， 又该给多少权重； 一般付出的代价越高， 人也就越在乎这种物品， 所以我们根据用户的付出代价的多少来判断给定行为特征权重。 相反， 在网站中只是看了一眼物品， 那么这个物品对用户来说并不显得有多重要。 那么对应的权重也相对较少。

用户行为产生的时间 人的兴趣有长期和短期两种， 但是不管是长期还是短期近期产生的权重较高的一定是用户感兴趣的。 比如一个23岁的用户最近一直在看汽车方面的内容， 并预约去试驾某类型车型，那么有很大的可能就是该用户又买车的意愿了。 最近的预约行为对应的权重就相对较高。

用户行为的次数 正如上面那个例子用户在很长一段时间内一直在观看与汽车有关的新闻，对于汽车这个关键字用户产生了多次行为。 再比如用户会听一首歌很多次，看一部电视剧的很多集等。 由此可得出以下结论：{42％： 用户对某种特征产生了多次行为，行为的次数越多代表用户对此越有兴趣，那么权重也就越高。}

物品的热门程度 热门的物品往往有很多人关注， 并非一定对此物品感兴趣， 只能证明商家对该物品的广告力度强大。 换句话来说就是： {59％：对热门物品产生行为不能代表用户的喜好。} 因为用户可能是在跟风，可能对该物品并没有太大兴趣，特别是在用户对一个热门物品产生了偶尔几次不重要的行为（比如浏览行为）时， 就更说明用户对这个物品可能没有什么兴趣，可能只是因为这个物品的链接到处都是， 很容易点到而已。但是假若用户对一个比较冷门的物品产生了行为，那么很有可能用户对物品缺失感兴趣， 从用户兴趣的长尾理论也可以推算出该结论。 我们可以对这个冷门物品增加权重。

特征-物品相关推荐

{64％：在离线相关表中根据该用户的特征向量我们可以得到物品的初始推荐列表，离线相关表可以存储在MySQL中。}

{46％：表 4 1特征-物品相关推荐表}

Src_idReco_idItem_idWeightAdd_time

{45％：特征ID算法ID物品ID权重时间戳}

在线使用的特征? 物品相关表一般都不止一张。 以论文之间的相关表为例， 计算论文之间的相关性既可以使用第3章提出的协同过滤算法（即如果两篇论文的读者重合度很大说明两部电视剧相似）， 也可以通过内容计算（比如有相同的作者、关键词、相似的标题等）。 {40％：即使是协同过滤，也可以根据不同的用户行为数据得到不同的相关表。} {42％：比如可以根据用户的打分行为计算论文之间的相关性，也可以根据用户的浏览行为计算论文之间的相关性。} 总的来说， 推荐系统存在很多配置文件，其中配置了相关表的权重以及推荐策略。 当系统启动之后，这些皮遏制会被载入系统，推荐时直接引用这些配置。

每一个推荐列表都需要一个合理的解释， 这个解释就是推荐理由。 只有正确的理由才能合理的解释推荐这个物品列表的原因， 也可说明这些物品是哪些特征产生的。

过滤模块

{47％：初步的推荐列表并不代表可以最终呈现在用户面前的推荐列表。} 虽然它符合了用户的特征向量，但是还存在一些常识或者不符合的物品。 {41％：我们需要通过以下几点对列表进行处理。}

{45％：用户已经产生行为的物品推荐系统是为用户发现用户会感兴趣的物品，并不是用户已经满足了的兴趣点，} 比如已经产生过购买行为的物品再推荐的话，用户再次产生行为的概率极低。 同时也可以保证推荐的新奇性。

候选物品以外的物品 候选物品集合一般有两个来源，一个是产品需求。 比如在首页可能要求将新加入的物品推荐给用户，因此需要在过滤模块中过滤掉不满足这一条件的物品。 另一方面推荐列表中产生了用户感兴趣的

列表之后，用户同时有明确了几个新的条件比如按销量，或者发布时间等这些条件，那么不符合这些条件的物品也要过滤掉。

某些质量很差的物品 这个大家都比较理解，货真价实才是王道。 推荐也要求货真价实。 {44%：那些评价很低的物品被加入推荐列表严重影响用户对推荐系统的信心。}

排名模块

为了极致的推荐效果，我们可以对推荐结果做进一步的优化，也就是排名，比如销售量，价格等。 {48%：一般排名模块需要具有代表性的子模块，下面对这些模块进行一一介绍。}

时间多样性

不同的时间地点产生不同结果，推荐结果也是如此，用户每次看到的推荐列表总是一成不变的，即使再好的物品也会变的即为无趣，这是视觉和精神上的双重疲劳。 {45%：提高推荐系统的时间多样性要从以下几个方面着手。} 一方面保证系统的实时性。 {57%：用户新的行为要实时作用于推荐列表。} 若没有新行为那么可以通过以下方式来实现推荐列表的多样性： 推荐结果随着时间的变化排名按一定的顺序变化。 再者将这些结果发回日志系统，记录结果日志，根据一定的时间将推荐结果降权，来优化推荐策略。

用户反馈

{43%：提高推荐结果的精确度最好的方式就是用户对推荐结果的反馈数据，这些数据反映了用户对推荐列表中物品的态度。} {49%：一般我们通过点击模型 来预测用户的未来点击行为，从而提高推荐的相关性。} 当然这个模型在各种点击预测中有广泛的应用，比如广告的点击预测，推荐列表上下文点击预测等等。 {49%：总之，用户反馈数据对于推荐结果的作用是明显的。}

{87%：系统非功能性需求描述}

实时性； 前面将用户兴趣进行了分类，在兴趣类别中特别是短期兴趣是不断发生变化的，若系统不能做到实时那么系统将会出现推荐结果不可靠，或者并不是用户感兴趣的。

准确性； 这个主要依赖于推荐算法的选择和对用户具体场景的判断。 所以推荐排名的策略就应该更加智能。

VIEW系统概述与分析

{43%：本系统的主要功能是推荐单子的展示，用户行为数据的收集；} 推荐单子的展示不仅仅以网页的形式展现，而且提供web service服务，RSS推送等功能；

用户行为收集

推荐只有拿到了用户的行为数据，才可能进行推荐，而用户行为从反馈的种类可分为显性和隐形，显性数据一般能明确表达用户的兴趣，需要持久化； 而隐形数据表达用户喜好程度较弱，一般在内存中存储。 如当天在站内浏览物品的信息，推荐结果只需要知道你浏览了什么，然后根据浏览的物品即可做出推荐。 {45%：所以

隐形数据在现阶段不做存储。}

推荐理由/推荐结果生成

每一种推荐都需要一个推荐的理由，也可以说没有推荐理由的推荐不是好推荐。这里暂时提供两种：1.获取推荐的具体好友信息，2.通过历史浏览记录得到的推荐结果或者理由。

远程服务

为第三方接入推荐列表提供接口服务，第三方主要是推荐结果的展示，和推荐结果反馈信息的收集。获取反馈信息分析和产出推荐规则，如IT行业的从业人员喜欢购买计算机类书。然后根据规则精确推荐结果。

安全服务

用户信息在传输过程中很容易受到攻击，为了保证用户信息的安全，在部分用户信息传输上会采用https协议进行传输；{45%：当然在采集用户的行为数据之前，肯定是取得用户同意的。}

日志系统概述与需求分析

现阶段比较流行的网站一般都有以下特性：{44%：访问量大，网站结构复杂，规模大等；}而且性能要求比较高，而一般用到推荐的系统往往是信息的规模达到了一个量级，仅仅几个页面不能再满足给所有用户展示所需要的信息了。{52%：一个完善的用户收集系统，是必不可少的一部分。}它可以帮助你监控网站流量，系统是否正常，用户的行为变化等等信息，我们主要是对用户行为的监控，分析，利用。当然系统级别的也是需要的，而且也有必要，保证网络的正常，系统出错提醒等。

日志系统需求分析

{47%：日志系统的任务是收集用户的行为日志，跟踪用户行为，清洗用户行为数据。}在可靠的日志系统中，用户的行为是保证成功记录下来的，不会出现遗漏的情况。日志系统可分为两大类在内容方面，一是业务日志，也就是本系统的主要功能，为推荐引擎提供准确有效的用户行为数据，从日志数据中分析具体的业务；二是系统日志，跟踪系统运行情况，若出现错误根据日志内容分析错误原因。

{40%：用户的行为日志量中不但有显性反馈数据，而且还有用户的隐形数据，比如用户点击物品的次数，}在站内停留时间等，这些数据造成日志系统的数据输出量成倍增加，从而影响整个系统的性能。如果从数据价值方面考虑的话，并不是所有的数据都是有价值的，而这些数据也增加日志信息检索的困难。

为了提供更好的日志服务可以从以下几方面进行考虑：

数据便于检索，可以通过SQL或者数据查询语言快速计算所需数据；

{43%：用户行为日志数据层次清晰，组织合理。}

可控可扩展，允许用户动态修改，特别是线上生产情况下。比如新增统计某个信息提交的次数。

日志系统构成

{42%：根据具体的业务需求，日志系统主要分为以下几块：} {41%：用户行为来源，数据流程管理，数据格式化输出，数据持久化；}

用户行为来源

用户的行为数据可以网页中任何元素，任何内容，也可以是系统中任何页面，这里面有各种类型的用户数据，比如点击，评价，浏览次数等等，如何有效的组织这些数据，这就需要一种规则来规范数据的形式了，如我们业务仅仅需要你浏览了某一商品的次数，那么系统就收集次数。其本质就是日志系统制定规则来规范数据，过滤数据。{58%：以便系统对数据进行管理和使用。}

数据流程管理

业务需要一件商品一天内的用户浏览量，这是我们就要控制系统在收集该类数据的时候要控制时间是24小时。数据流程管理的主要工作就是控制数据收集的时间，地点，收集的格式等等。其本质就是控制数据在系统内的执行过程。类似于Linux的任务管理。

数据格式化输出

最常见的输出方式就是从控制台输出了，无论我们学习哪种语言，第一节课的内容肯定是在控制台输出 Hello World，但是生产中的输出，不仅仅是控制台输出，当然在调试的时候我们也是需要控制台输出的，但是如果我们不控制信息的级别那么 info类型的，warning类型的，error类型的都输出到控制台，对于我们 debug也是不利的。格式化的本质就是将数据按一定的方式组织在一起，以方便用户的存储，分析等。

数据持久化

{40%：用户数据根据业务的不同，对持久化的要求也是不同的，有些数据组织形似方便存储在关系型数据库中，} 而JSON类型的数据就不那么的适合了，这时候就会选择 NOSQL类型的数据库了，比如 MongoDB；还有就是考虑数据的大小，还有是否实时这些因素，这时就要用到内存数据库或者缓存了。

日志内容

内容分级

{40%：日志系统输出的数据不仅仅包含用户行为日志，还有系统日志，应用程序日志等，} 若想更好的管理这些日志内容，我们需要将数据进行分级，通过一个全局常量来控制，如果日志输出级别是用户行为级别，那么记录，否则放弃忽略不计。从而来控制数据的输出。也可以根据日志级别配置日志输出路径，从而将数据分流。

用户行为分类

当用户行为数据多到一定程度的时候就需要将其分类管理了，否则随着数据量的增长，查询所消耗的时间也会增加，进而影响系统的正常运行，从用户行为主动或被动角度来考虑，{58%：可将数据分为显性与隐形反馈

数据；} {44%： 从用户对物品感兴趣与否可将其分为正反馈， 负反馈；} {43%： 从数据的产生途径可分为元数据， 中间值， 结果这三类。}

内容格式

之前我们对内容进行了分级和分类， 但是对于用户行为数据来说， 粒度还是比较粗糙。 我们还要根据业务需求来更加详细的划分内容， 因为只有大量的用户行为数据才有统计的意义， 但是大量的无用数据会造成极大的系统负担。 这对于系统来说是致命的。 {43%： 还有就是数据的多样性， 我们可以通过以下方式来解决：}

用XML格式来存储， 这样可以解决数据多样性的问题， 但是同时也带来了解析数据时吓人的时间复杂度。 在这通过JSON 来存储这些数据， 不仅方便解析， 而且现在很多网站在网际之间使用该格式来传递消息的。 如果以上还是不能满足需求那么就根据具体的业务设计一个新的数据格式来处理这种数据，一般前两种方式就可以满足大部分需求了。

日志系统目标

易用性

{50%： 易用性，是每一个软件都在追求的特性。} 易用的目的是为人提高效率。 我们使用计算机的根本原因就是它能够让很多工作变得简单以及为我们节省时间，否则它就没有被我们使用的价值了。 我们可以这样说，大多软件特性都是易用性的延伸，比如，国际化、扩展性以及高效性等。

性能

推荐系统如果需要实时推荐的时候， 那么系统的相应时间不能超过3秒， 如果纯粹从系统性能的方面来考虑，日志是不应该存在的。 但是对于推荐系统那是不可能的， 当打开所有任务，系统的性能将会受到极大的影响，这时候就体现出了给日志内容分级显的尤为重要了， 我们一般只开启部分任务， 当然我们也可以制定一个异步任务。

国际化

如果仅仅是英文或者数字字段， 不进行国际化也是可以的， 但是有时候针对评论字段， 那就不能不国际化，考虑国内情况； 还是需要进行国际化的。 {41%： 当然也不需要全部的国际化， 仅仅将用户行为数据国际化即可。}

本章小结

本章主要是对推荐系统整体进行需求分析， 并将整体分割为三个子系统，然后对各个子系统进行详细的需求分割， 划分功能模块。 以下是各个子系统的描述： 推荐引擎， 在本节中将推荐算法划分为最小推荐模块，推荐算法有很多， 同时推荐的需求也不断的变化， 划分推荐引擎为最小单元有利于算法和业务的扩展。 {41%： VIEW系统主要是划分展示的形式， 以及定义各种功能接口。} 日志系统在于制定日志系统的大体规范， 以保证高效获取推荐所需的各类数据。

推荐系统设计与实现

{48%：推荐引擎总体架构和J2EE技术概述}

架构目标

本系统是为推荐系统的核心，而且兼容多种推荐算法，因此综合考虑本系统的设计目标如下：

性能要求。 要求系统能够提供良好的运行性能，能够满足大量用户的并发，并且要保证每个用户的私有信息得到保护。

系统可维护性： {44%：为了适应系统的功能扩展，系统的架构必须适合扩展，符合“开-闭”原则；} 在系统的开发过程中，系统文档、指示代码、体系应遵循软件开发的规范要求，使系统具有良好的可维护性。

整体设计框架

{62%：图5 1推荐子系统流程图}

推荐引擎功能设计

用户特征模块

{43%：本模块主要功能就是获取用户特征和权值；} {43%：通过用户的ID，或者用户的属性字段，可以是单个或者多个属性字段。} 如性别，年龄等的组合条件。

{46%：得到人口统计学中的行为标签}

根据用户的ID从用户注册信息表中抓取用户的特征，和用户特征的权值，这个权值可通过枚举类进行配置。

计算用户行为特征

根据用户行为的种类，产生时间（可选），行为的次数（可选），热门程度（可选）作为传入参数考虑，返回用户的行为记录，然后计算用户的行为特征；

推荐引擎结构图

推荐算法有很多，而且在实际生产中并不是一个推荐算法就可以满足所有的业务， 以持续发展的眼光来看待事物的话，这个系统是不断扩展的，将各种特征什么的都写入一个配置中， 那么后期配置会越来越麻烦最终将无法再修改。 最好的解决方案就是单一责任制原则，即一个推荐引擎负责一个任务，一类特征。 {68%：而推荐结果只是按照一定权重或者优先级合并，排序即可。}

图5 2推荐引擎结构图

过滤和排名模块

根据商品的打分情况， 还有用户是否购买过此商品来过滤最初的推荐列表， 然后对列表进行排名；

View系统总体技术架构与设计

{45%：本系统应用三层软件架构模式，其中包含表示层、业务逻辑层和数据访问层，表示层主要负责与用户的交互，即将得到的数据显示出来；} {57%：业务逻辑层主要负责处理业务提供相应的业务接口；} {56%：数据库访问层主要负责从数据库提取所需数据。} {40%：三层结构各司其职，系统层次感分明，很好地降低了系统的耦合度，提高程序后续扩展的简便性。}

下面是本系统的整体框架图：

图 5 3视图系统的总体架构

功能模块分布

图 5 4VIEW系统功能模块

用户行为模块

通过推荐界面提供一些按钮或者在一些特殊的图片，按钮， 链接上加入监控收集用户行为数据并让用户对推荐结果进行反馈，这样才能让推荐算法不断改善用户的个性化推荐体验。 以下是用户行为接口设计:

图 5 5用户行为包设计

显性行为数据收集

程序描述：

用户购买行为收集： 用户在产生购买行为时用户的行为收集， 主要包括用户购买商品的价格， 数量， 商品的类别，商品的种类等信息。

{60%：用户查看推荐物品行为收集：} {52%： 用户点击推荐栏中商品所产生的信息；}

{45%：销售排行信息产生的用户行为：} {44%： 热门排行， 销售排行， 新品推荐等用户行为收集。}

用户主动评分信息的收集： {49%： 用户主动为物品评的分数， 评价等信息；}

隐形行为数据收集

浏览商品用户行为收集： 用户在展示页面中浏览商品， 系统收集用户在浏览器中浏览商品的时间， 商品信息， 浏览地区等信息并存储到日志系统内；

站内搜索关键字收集： {40%： 用户在站内搜索框搜索站内内容， 系统收集用户的内容和时间。}

最近查看的商品的行为： {44%：用户在一段时间内所浏览的商品信息，也可以说浏览商品详情页的信息。}

用户行为调查报告收集

主要提供的功能是为一些特殊物品或者用户行为信息收集提供的接口，如对MacBook的认识，价格，性能等的调查报告。

调查问卷的生成： 为特殊的需求生成调查问卷；

调查信息的收集： 收集用户调查报告的数据；

推荐结果/推荐理由展示

系统根据具体的条件如： 推荐物品的数量，推荐物品信息的选择，推荐物品更新时间间隔等因素从系统中返回推荐结果，并附加推荐解释，也就是推荐理由。

获取浏览历史推荐结果： 根据ID（mac地址/用户ID）监听用户在网站中对物品的浏览记录，并返回。

获取销售排行推荐结果： 根据物品的销售量将物品进行排序并输出排名最高的N件物品。

获取物品组合推荐： 物品详情页中获取与物品相关的物品，加入组合推荐栏目中。

远程接口与用户安全设计

远程接口

本系统提供在应用层的远程调用，采用HTTP协议进行系统间的传输，消息之间的传输机制有两种，分别是同步传输和异步传输。 {46%：而系统作为消息的生产者在生产消息的时候为消息加上了消息的控制机制。} 每个注册消费机只能有一次消费消息的机会。 而这些接口的设计应符合以下这些条件：

通信技术: {46%：通信我们采用Netty技术，通过此框架快速构建服务端与客户端之间的通信协议。} Netty是当下比较流行的通信框架，而且性能相对于Java 同步阻塞（BIO）高出许多。 同时Netty支持多种协议，完全满足系统的需求。

{78%：序列化和反序列化技术：} 本系统我们采用的是Java语言来实现的，一次我们所使用的也是Java自带的序列化技术，序列化主要的目的就是为了是Java对象能够远程传输； 将Java对象从服务器传到客户端，其实质就是讲Java对象以固定的算法转化为字节流。 这些字节流可以方便自由的网络中传输。 即便是跨系统也不会影响。 {48%：将对象转化为了字节流，同样的也要有办法将字节流转化为对象，而这种技术就是反序列化。} 反序列化其实质就是对字节流进行解析按照一定的方式。 Java性能总是有那么一些不尽人意的地方，比如说序列化，又有对象的臃肿， 而序列化的时候将这些信息也进行了序列化，最终是得系统的序列化效率低下， 影响字节流在网络中的传输速度，而且解析麻烦的情况。 但是由于本系统中主要使用序列化的地方在远程调用方面，而远程调用在系统的占比并不大，从而整体对系统的影响可以忽略。 当然，如果可以的话最好预留相应的扩展接口

以便将来扩展相应的需求。

压缩技术： 但凡需要系统设计网络方面的编程，那么压缩技术就是必不可少的一部分，为了节约网络资源，我们将一些图片，文本等这些静态资源同时也包括代码要进行压缩，本系统才有的压缩技术是：{43%：对序列化的压缩我们采用thrift压缩二进制编码技术。} 而静态资源的压缩采用最常用的Tomcat的GZIP压缩技术。

非侵入式： 在web开发中代码大体可分为业务代码和系统代码；而系统代码中往往要请入到业务中去，比如Log日志，监控业务的执行流程。 我们在方法前需要打印方法开始执行，方法执行中我们需要打印方法中间结果到日志中区，方法结束的时候我们需要关闭资源等等。 这些系统代码将业务代码污染了，而自身也无法保证质量。 {42%：因此我们使用Spring框架中的AOP编程，它使得业务和系统之间相分离。} 这样代码的实例全部由Spring容器进行管理，而对象以对象属性的方式或者构造方法传入对象中即可，大大的将业务和系统解耦。 {42%：这种非侵入式的设计也给测试带来了福音。}

高并发的技术： WEB系统随着业务的增加系统IO之间的负荷也越来越高，如何提高性能。 高并发技术是必不可少的，单台WEB服务器与数据库之间的连接数大概在500左右，当QPS超过该量级的时候就会造成系统崩溃。 {48%：我们这里才有C3P0线程池来解决数据库连接方面的问题。} 接着访问连接池我们使用Apache的common-logging来实现。

负载均衡： 负载均衡在分布式系统中是必不可少的一部分，它是解决高访问量下解决系统负载问题的中坚力量。 {41%：还有我们在产品上线的过程中如何保证系统能够平滑的切换。} 当出现恶意请求的时候负载均衡可以挡住这些请求不让这些请求进入系统内部。 本机暂时采用轮询机制来进行服务器的访问，当我们需要上生产的时候， {51%：我们会将一台服务器从线上摘取下来，所有外部请求无法到达该服务器，} {44%：然后将系统部署到服务器之后负载均衡再将该服务器接入系统。} {47%：多台服务器可以呈梯度趋势进行部署上线。} {44%：这样保证了系统在在线的情况下进行系统的升级。}

用户安全

系统中涉及用户信息安全的主要在用户信息传输方面，这时候需要合理的机制来保证信息的机密性， {42%：还是防止在网络中出现丢包的现象，验证信息的完整性和真实性。} 在信息安全领域，避免系统中出现多个与外界交互的接口，而且每个接口都需要完整的安全验证。 其中涉及的技术如下：

加解密技术。 {43%：我们采用非对称加密和对称加密结合的方式来保证数据的安全性，我们首先生成一对分对称密钥，将公共密钥传输给目标主机，} {45%：目标主机在生成一个对称密钥，并将对称密钥通过分对称公钥进行加密传给源主机。} 之后两系之间通过对称密钥传输信息。 保证信息安全传输。

VPN技术。 系统中只有在第三方配置推荐物品清单，或者接口系统后台管理的时候，使用企业提供的VPN连接系统，通过在公网建立一条安全的路径。 从公网接入系统内容进行数据操作。 {41%：本系统通过租用电信的MPLS VPN专线服务，从而降低开发的成本。}

防火墙技术。 就好比在院墙一样，院墙是将家和大街通过一堵墙隔开；{42%：同理防火墙是将外网和内容隔离开。} 当人要进入院内的时候必须通过院的主人验证是否为合法的人才允许进入。 而外部网络要想访问内网，那么必须通过合法验证之后，才允许访问。 还有就是在墙内安置入侵检测系统，分析检测外部访问信息，检查是否违反安全规范并防止Ddos攻击。 {49%：本系统主要采用应用层的XML防火墙。}

安全审计技术。 包含日志审计和行为审计。 监控日志中Warning级别的日志信息，并在出现百量级的报错日志，发出邮件提醒网络管理人员，及时排查原因，评估安全机制。 再有记录系统中所有操作人员的所有操作日志，在执行越界操作的时候系统自动发出经过处理。 从而确保系统安全。

本章小结

{42%：本章主要内容是整个推荐系统的设计实现，其中包含推荐引擎，推荐VIEW系统的设计。} 推荐引擎技术是一种新的信息传播方式，将推荐引擎分为两部分一部分是推荐算法模块，另一部分属于规则模块，其中有热门排行，还有一些规则组成； {49%：两部分互补组成推荐的主要部分，再经过过滤和排名得到最终的推荐列表。} {41%：当然主要是解决推荐引擎的整体架构与技术架构实现。} 再有就是对VIEW系统的详细设计，其中主要包含是系统的总体技术架构，系统功能的描述，分析，具体接口的定义等，其中有用户行为模块，推荐结果模块，远程接口与用户安全设计。

总结与展望

{40%：推荐系统是解决信息过载的一种解决方案，同时也是一种连接用户和内容的信息系统，一方面它帮助用户发现他们潜在的兴趣点，} 另一方面它能够帮助信息提供者将内容投放给对相应的用户。 {50%：推荐系统的常用方法是通过分析用户的历史行为来发现他们将来的行为。} 但是在国内并没有一个能够为行业提供推荐的独立推荐公司，或者推荐系统； 百度在广告方面的推荐具有一定的优势，但不提供对外服务。 {40%：因此本文主要是推荐系统的设计与实践，为提供独立的推荐系统做出一点力量。}

推荐系统总结

{47%：本系统的工作包括一下两方面：}

通过SVD方法计算用户的与测评分，又使用Top-N推荐对隐形反馈数据计算用户-物品的权值， 返回用户的推荐列表，并在其原有的算法中加入一个新的时间元素，形成一个合适的算法。

{45%：对推荐引擎分析，日志系统以及展现系统分别进行了分析和设计；} 从使用的相关技术到系统的技术架构，再到相关接口的设计，每一点都做了详细和深入的探索。

推荐系统展望

本文对Top-N做了一定的优化，并对整个推荐系统的实践做了相应的阐述，但是针对用户的组合推荐和多特征并行处理的推荐并没有相应的实践。 {43%：希望在以后的实践中对推荐算法做进一步的改进，提高推荐算法的计算性能，算法推荐的精准度以及线上的相应时间。}