大数据应用及安全

推荐系统

提纲

1. 推荐系统
2. Youtube视频推荐

推荐系统

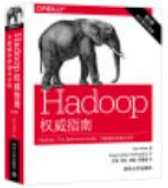
京东搜索《机器学习》

商品介绍　售后保障　商品评价(11万+)　加

C++ Primer Plus（第6版

33000条 (98%好评)

¥80.90 [8.2折]



Hadoop权威指南：大数据

22000条 (98%好评)

¥122.70 [8.3折]

高性能MySQL（第3版）

39000条 (98%好评)

¥89.40 [7折]

码农翻身：用故事给技术

53000条 (99%好评)

¥57.80 [8.4折]

智能时代 大数据与智能革

100000条 (99%好评)

¥48.10 [7.1折]

热门关注

算法竞赛入门经典（第2

19000条 (99%好评)

¥35.60 [7.2折]

编程珠玑（第2版 修订

69000条 (99%好评)

¥27.50 [7.1折]

白话深度学习与TensorFlo

23000条 (99%好评)

¥53.60 [7.8折]

人类简史：从动物到上帝

320000条 (99%好评)

¥68.00 [10折]

数字图像处理（第三版）

5800条 (99%好评)

¥66.30 [7.5折]

推荐系统

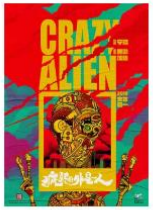
天猫搜索 足球服

推荐系统

豆瓣搜索 电影-无名之辈

喜欢这部电影的人也喜欢 · · · · · ·


天气预爆


疯狂的外星人


人间·喜剧


日不落酒店


情圣2


八个女人一台戏


枪炮腰花


唐人街探案3


素人特工


憨豆特工3

无名之辈的短评 · · · · · · (全部 64984 条)

我要写短评

热门 / 最新 / 好友

ლ♪生活有这些期待很有动力♫♫ლ (freedom♪)

豆瓣上评分人数超过5万评分高于7分的电影 (stupidliar)

谁在看这部电影 · · · · · ·


长乐无极
刚刚 看过 ★★★☆☆

201227人看过 / 145870人想看

订阅无名之辈的评论:
feed: rss 2.0

推荐系统

百度搜索 人工智能

为什么会出现推荐系统

信息超载

互联网的出现和普及给用户带来了大量的信息，满足了用户在信息时代对信息的需求，但随着网上信息量的大幅增长，用户在面对大量信息时无法获得对自己真正有用的部分，对信息的使用效率反而降低了，这就是所谓的**信息超载**问题

为什么会出现推荐系统

搜索引擎

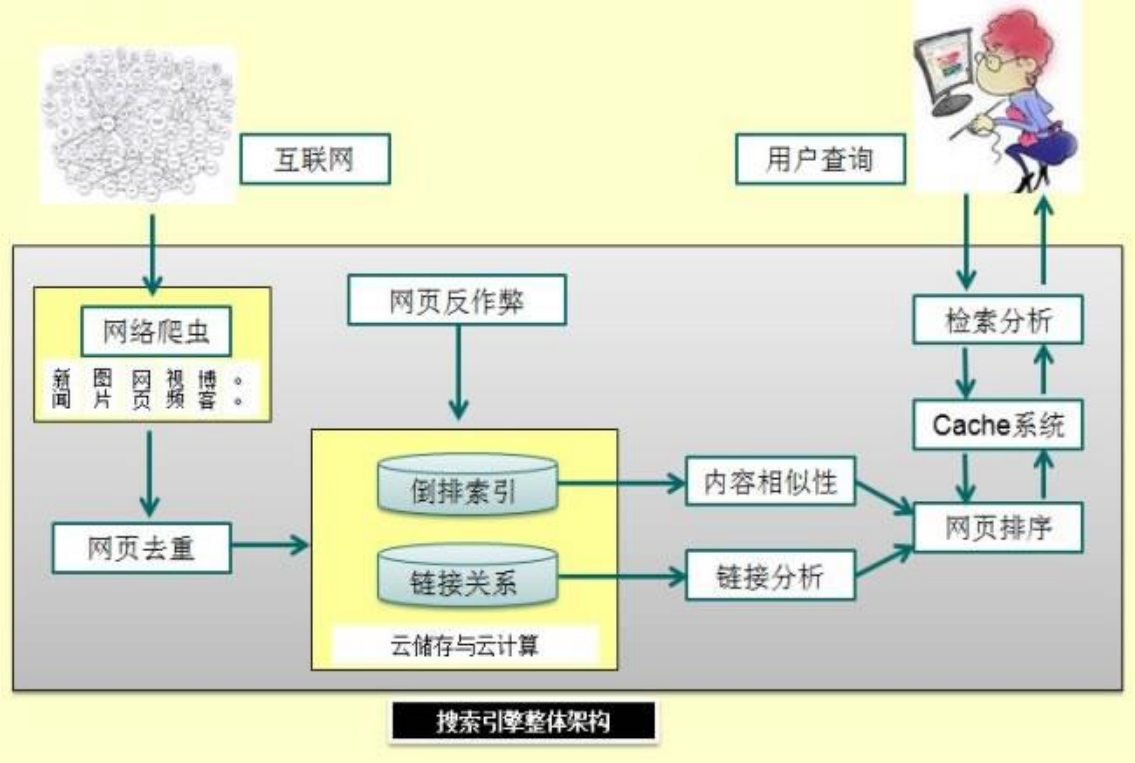
现有的很多网络应用，如门户网站、搜索引擎和专业数据索引本质上都是帮助用户过滤信息的手段。然而这些工具只满足主流需求，没有个性化的考虑，仍然无法很好地解决信息超载的问题

为什么会出现推荐系统

搜索引擎

用户在搜索互联网中的信息时，需要在搜索引擎中输入“查询关键词”，搜索引擎根据用户的输入，在系统后台进行信息匹配，将与用户查询相关的信息展示给用户

但是，**若用户无法想到准确描述自己需求的关键词**，搜索引擎就变得无能为力



为什么会出现推荐系统

推荐系统

推荐系统不需要用户提供明确的需求，而是通过分析用户的历史行为来对用户的兴趣进行建模，从而主动给用户推荐可能满足他们兴趣和需求的信息

推荐系统vs.搜索引擎

搜索引擎	推荐系统
注重搜索结果（如网页）之间的关系和排序	还研究用户模型（user profile）和用户的喜好，基于社会网络(social network)进行个性化的计算（personalization）
由用户主导，包括输入查询词和选择结果，结果不好用户会修改查询再次搜索	由系统主导用户的浏览顺序，引导用户发现需要的结果。高质量的推荐系统会使用户对该系统产生依赖
	推荐系统不仅能够为用户提供个性化的服务，而且能够与用户建立长期稳定的关系，提高用户忠诚度，防止用户流失。

推荐系统历史

- 推荐系统这个概念是1995年在美国人工智能协会（AAAI）上提出的。当时CMU大学的教授Robert Armstrong提出了这个概念，并推出了推荐系统的原型系统——Web Watcher。在同一个会议上，美国斯坦福大学的Marko Balabanovic等人推出了个性化推荐系统LIRA1
- 1996年， Yahoo网站推出了个性化入口MyYahoo，可以看作第一个正式商用的推荐系统
- 21世纪以来，推荐系统的研究与应用随着电子商务的快速发展而异军突起，各大电子商务网站都部署了推荐系统，其中Amazon网站的推荐系统比较著名。有报告称， Amazon网站中35%的营业额来自于自身的推荐系统
- 2006年，美国的DVD租赁公司Netflix在网上公开设立了一个推荐算法竞赛——Netflix Prize。 Netflix公开了真实网站中的一部分数据，包含用户对电影的评分。Netflix竞赛有效地推动了学术界和产业界对推荐算法的研究，期间提出了很多有效的算法
- 近几年，随着社会化网络的发展，推荐系统在工业界广泛应用并且取得了显著进步。比较著名的推荐系统应用有：Amazon和淘宝网的电子商务推荐系统、Netflix和MovieLens的电影推荐系统、Youtube的视频推荐系统、豆瓣和Last.fm的音乐推荐系统、Google的新闻推荐系统以及Facebook和Twitter的好友推荐系统

推荐系统——输入

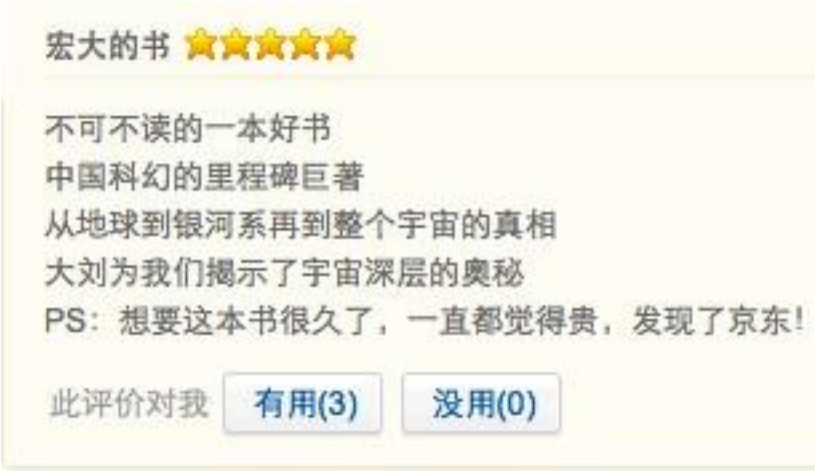
User + Item + Review

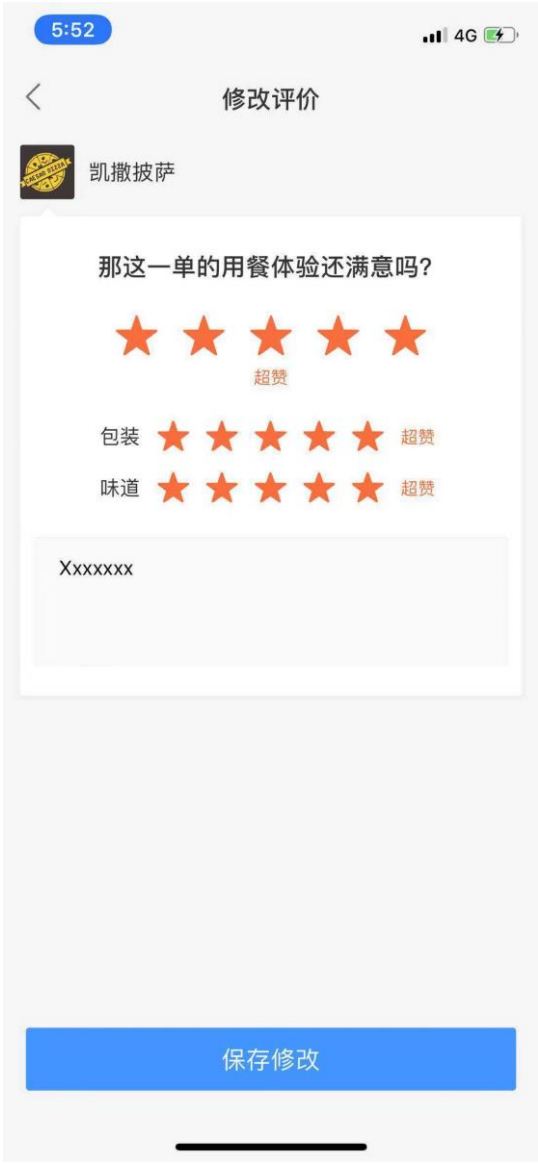
- **User & User Profile**
  - 描述一个user的“个性”
  - 两种构建User Profile的方式
    - 与Item Profile类似，如性别、年龄、国别、年收入、活跃时间……
      - ✓ 难以与Item建立具体的联系
      - ✓ 隐私问题
      - ✓ 很少直接使用
    - 利用Item Profile构建User Profile
      - ✓ Personalized IR related
- **Item & Item Profile**
  - 电影：类别、导演、主演、国家、。。。
  - 新闻：标题、本文、关键词、时间、。。。

推荐系统——输入

User + Item + Review

- **Review (user 对 item 的评价)**
  - **最简单的Review: 打分(Rating)**
    - **一般是1~5的星级**

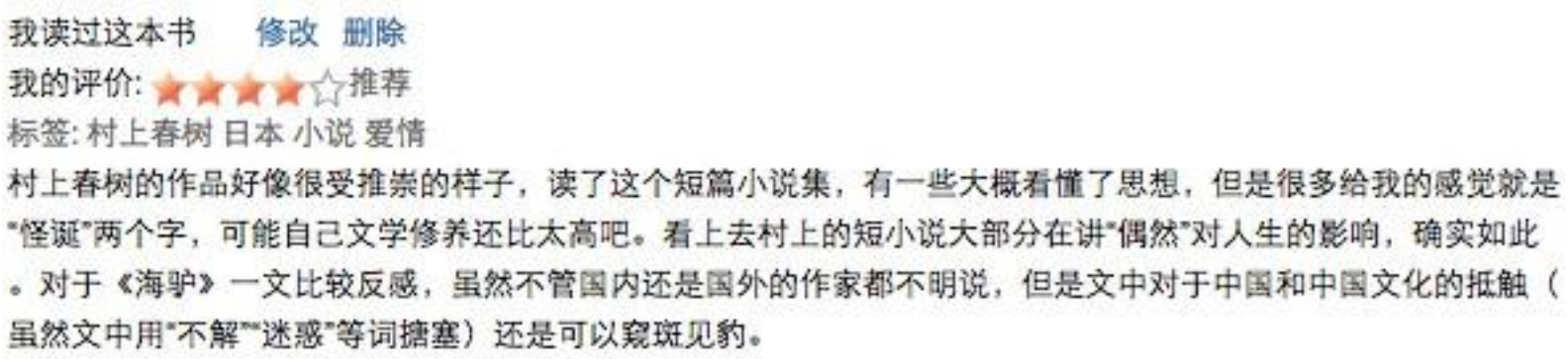




推荐系统——输入

User + Item + Review

- **Review (user 对 item 的评价)**
  - **最简单的Review: 打分(Rating)**
    - **一般是1~5的星级**
  - **其他Review**
    - **显式：评论、标签**
    - **隐式：查看历史记录、购买记录、页面停留时间**



推荐系统——输出

推荐列表

按照特定排序给出的对该用户的推荐

推荐理由

- 例如购买了A物品的用户有90%的人也购买了B物品
- 例如物品C在某类别中人气很高

衡量推荐的合理性

推荐系统——基本原理

用户行为数据

- 显性反馈行为
  - 用户明确表示了对物品喜好的行为
- 隐性反馈行为
  - 不能明确反应用户喜好的行为

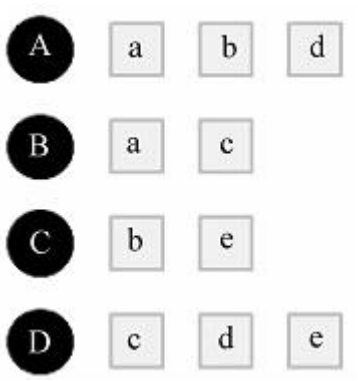
协同过滤算法

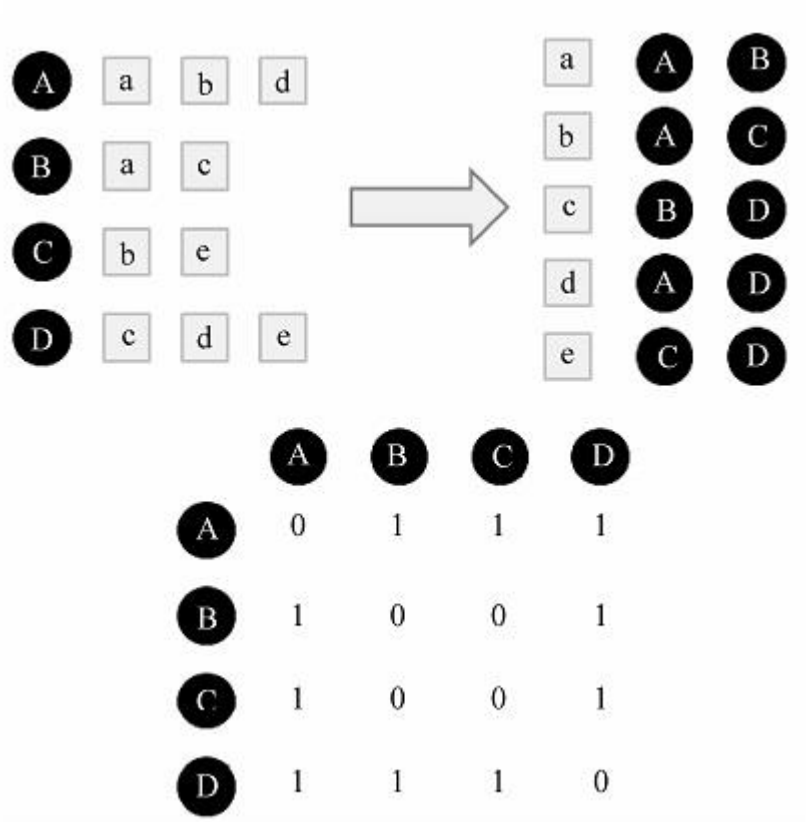
- **协同过滤**是指用户可以齐心协力，通过不断地和网站互动，使自己的推荐列表能够不断过滤掉自己不感兴趣的物品，从而越来越满足自己的需求
- **基于用户的协同过滤算法**(User-CF)：给用户推荐和他兴趣相似的其他用户喜欢的物品
- **基于物品的协同过滤算法**(Item-CF)：给用户推荐和他之前喜欢的物品相似的物品

推荐系统——基本原理

User-CF

先找到和他有相似兴趣的其他用户





余弦相似度公式

$$w_{uv}=\frac{|N(u)\cap N(v)|}{\sqrt{|N(u)||N(v)|}}$$

$$w_{AB}=\frac{|\{a,b,d\}\cap\{a,c\}|}{\sqrt{|\{a,b,d\}|\,|\{a,c\}|}}=\frac{1}{\sqrt{6}}$$

物品-用户倒排表

推荐系统——基本原理

User-CF

User-CF算法会给用户推荐和他兴趣最相近的K个用户喜欢的物品

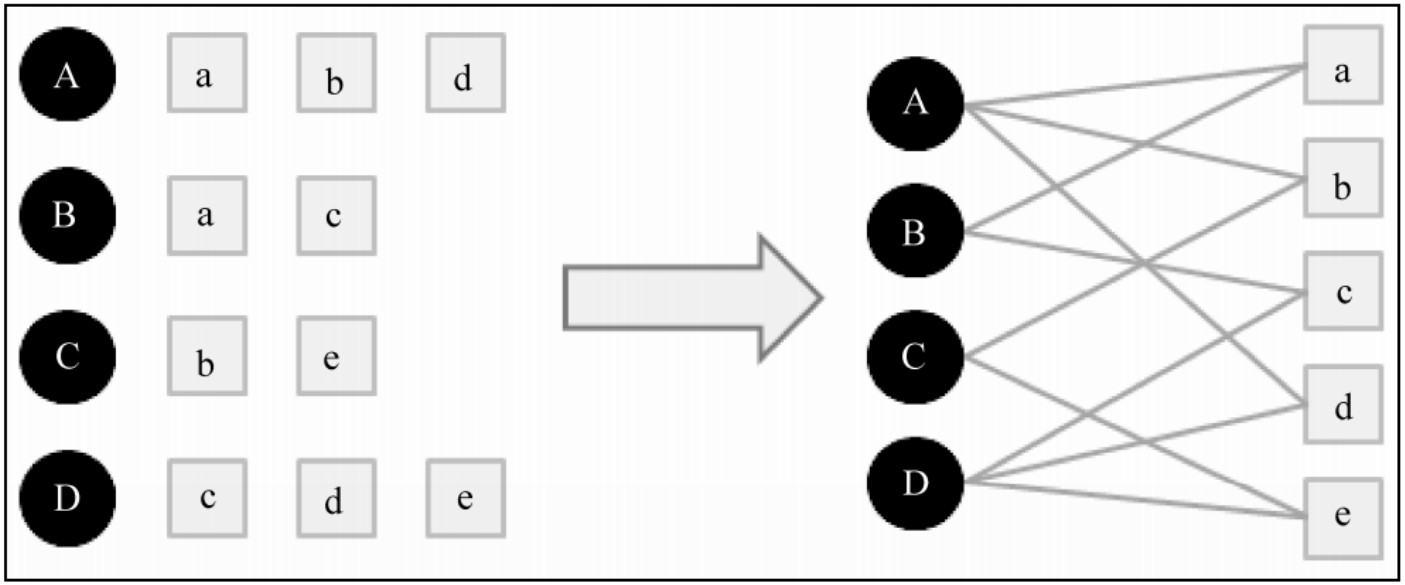
$$p(u,i) = \sum_{v \in S(u,K) \cap N(i)} w_{uv} r_{vi}$$

S(u, K): 包含和用户u兴趣最接近的K个用户
N(i): 对物品i有过行为的用户集合
Wuv: 用户u和v的兴趣相似度
Rvi: 代表用户v对物品i的兴趣

推荐系统——基本原理

基于图的推荐算法

- ✓ 二分图又称作二部图，是图论中的一种特殊模型
- ✓ 设G=(V,E)是一个无向图，如果顶点V可分割为两个互不相交的子集(A,B)，并且图中的每条边（i，j）所关联的两个顶点i和j分别属于这两个不同的顶点集(i in A,j in B)，则称图G为一个二分图。**用户行为很容易用二分图表示**，因此很多图的算法都可以用到推荐系统中



用户物品二分图模型

推荐系统——基本原理

基于用户标签数据

- **通过一些特征（feature）联系用户和物品，**给用户推荐那些具有用户喜欢的特征的物品

基于上下文信息

- 用户所处的上下文（context），包括**用户访问推荐系统的时间、地点、心情**等，对于提高推荐系统的推荐效果是非常重要的

基于社交网络信息

- **基于社交网络的推荐可以很好地模拟现实社会：**著名的第三方调查机构尼尔森调查了影响用户相信某个推荐的因素。调查结果显示，90%的用户相信朋友对他们的推荐，70%的用户相信网上其他用户对广告商品的评论

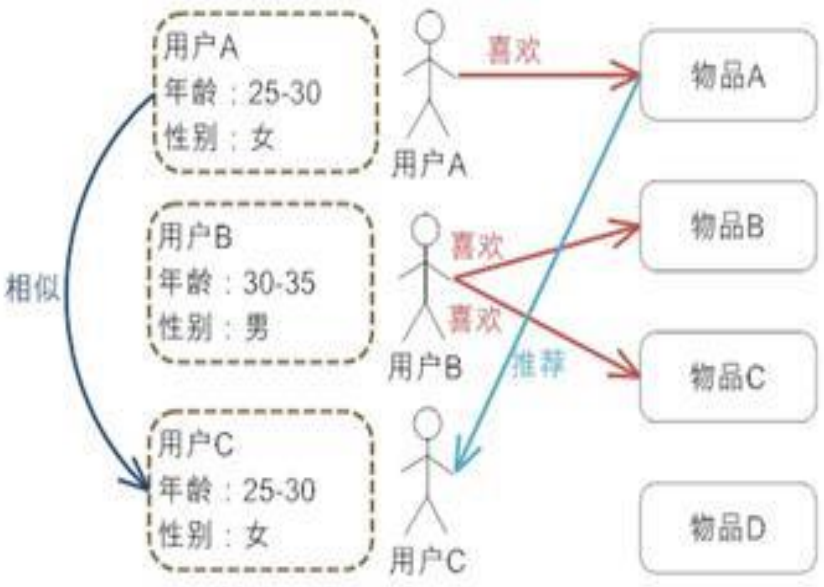
推荐系统——算法

- 基于人口统计学的推荐算法
- 基于内容的推荐算法
- 协同过滤推荐
  - 基于启发式的协同过滤算法
  - 基于模型的协同过滤算法
  - 基于图的协同过滤算法
- 混合推荐算法
- 基于关联规则的推荐算法
- 基于效用的推荐算法
- 基于知识的推荐算法

推荐系统——基于人口统计学

- 基于人口统计学的推荐算法是最为简单的一种推荐算法，只是简单的根据系统用户的基本信息发现用户的相关程度，然后将相似用户喜爱的其他物品推荐给当前用户

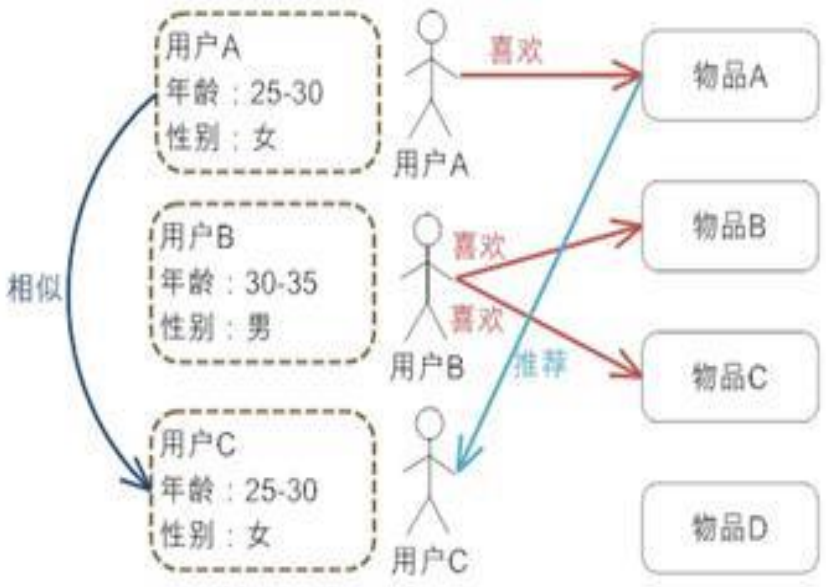
- 根据用户的属性建模，如年龄，性别，兴趣等信息。根据这些特征计算用户间的相似度。若系统通过计算发现用户A和C比较相似。就会把A喜欢的物品推荐给C



推荐系统——基于人口统计学

优点

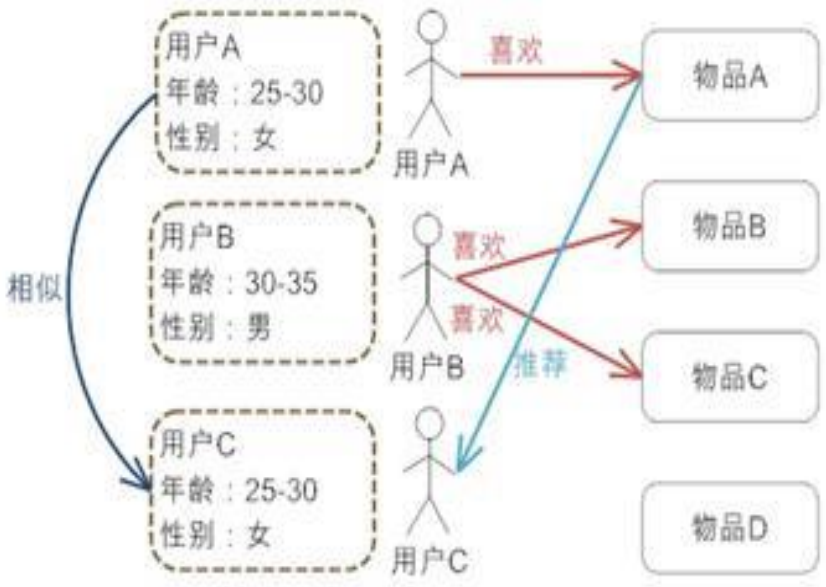
- **不需要历史数据：没有冷启动的问题**
- **无以来物品的属性**



推荐系统——基于人口统计学

缺点

- **算法比较简单，效果很难令人满意**



推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）

算法模型介绍

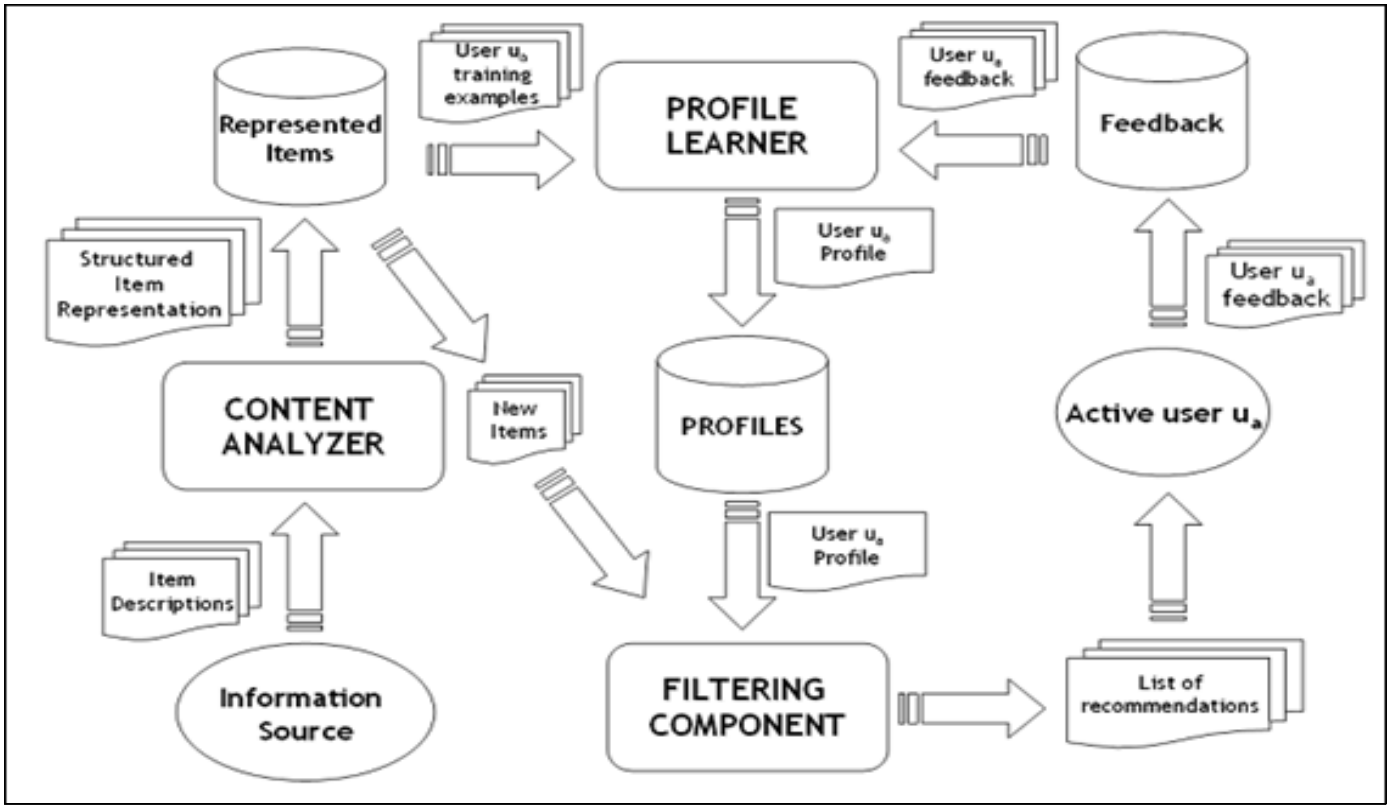
根据用户过去喜欢的产品（ item），为用户推荐和他过去喜欢的产品相似的产品。例如，一个推荐饭店的系统可以依据某个用户之前喜欢很多的烤肉店而为他推荐烤肉店

主要包括如下三个步骤

- Item Representation：为每个item抽取出一些特征,用来表示此item；
- Profile Learning：利用一个用户过去喜欢（及不喜欢）的item的特征数据，来学习出此用户的喜好特征（profile）；
- Recommendation Generation：通过比较上一步得到的用户profile与候选item的特征，为此用户推荐一组相关性最大的item。

推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）



CONTENT ANALYZER ----- Item Representation
PROFILE LEARNER ----- Profile Learning
FILTERING COMPONENT ----- Recommendation Generation

推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）

Item Representation

Item Representation：从Item中获取特征的步骤

- Item的属性可以分为**结构化属性**和**非结构化属性**两种，结构化的属性例如颜色、价格等可以直接当作特征；对于非结构化的属性例如Item的描述文本，需要先转化为结构化数据
- 对于文本类的非结构化数据，为了将其转化为结构化的数据，常用的办法有TF-IDF、词向量等方法。

✓ **TF-IDF**(即词频-逆向文件频率)是一种自动提取关键词的算法，通过该算法可以将文本转化为特征向量。

✓ **词频**（term frequency，tf）指的是某一个给定的词语在该文件中出现的频率

✓ **逆向文件频率**（inverse document frequency，idf）是一个词语普遍重要性的度量

推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）

Profile learning

Profile Learning：学习用户的偏好

- **K近邻算法**：对于一个新的item，K近邻方法首先找用户u已经评判过并与此新item最相似的k个item，然后依据用户u对这k个item的喜好程度来判断其对此新item的喜好程度

- **决策树算法**：当item的属性较少而且是结构化属性时，可以使用决策树算法来学习用户的喜好特征。这种情况下决策树可以产生简单直观、容易让人理解的结果。因为可以把决策树的决策过程展示给用户u，告诉他为什么这些item会被推荐

- **Rocchio算法**：基于用户的行为(例如点击行为)生成一个偏好向量，通过对比偏好向量和item向量的相似度来度量用户对于该item的喜爱程度

推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）

优点

- **用户之间的独立性（User Independence）**：每个用户的profile都是依据他本身对item的喜好获得的，与他人的行为无关。这种用户独立性带来的一个显著好处是别人不管对item如何作弊（比如利用多个账号把某个产品的排名刷上去）都不会影响到自己
- **可解释性强（Transparency）**：方便向用户解释为什么推荐了这些产品给他
- **新的item可以立刻得到推荐**（New Item Problem）：只要一个新item加进item库，它就马上可以被推荐，被推荐的机会和老的item是一致的

推荐系统——基于内容的协同过滤

基于内容的推荐算法（Content-based Recommendations）

缺点

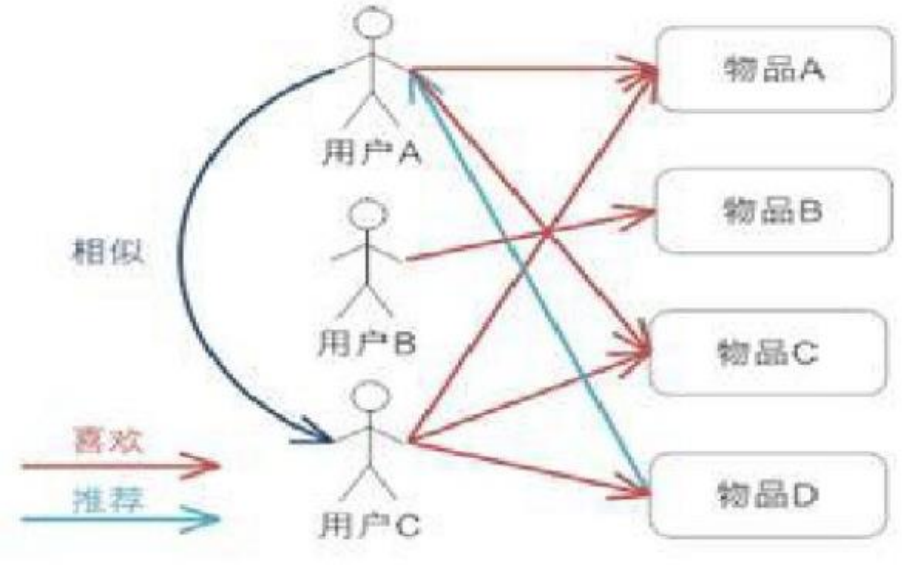
- **item的特征抽取一般很难**（Limited Content Analysis）：如果系统中的item是文档，可以比较容易地使用信息检索里的方法来抽取出item的特征。但很多情况下我们很难从item中抽取出准确刻画item的特征
- **无法挖掘出用户的潜在兴趣**（Over-specialization）：推荐只依赖于用户过去对某些item的喜好，它产生的推荐也都会和用户过去喜欢的item相似。如果一个人以前只看与推荐有关的文章，那只会给他推荐更多与推荐相关的文章，它不会知道用户可能还喜欢数码
- **无法为新用户产生推荐**（New User Problem）：新用户没有喜好历史，自然无法获得他的profile，所以也就无法为他产生推荐了

推荐系统——基于启发式的协同过滤

基于启发式的推荐算法（ collaborative filtering ）

基于用户的协同过滤

核心思想：基于用户对物品的偏好找到相邻的邻居用户，然后将相邻用户喜欢的物品推荐给当前用户



例如：老张喜欢看的书有A，B，C，D；老王喜欢看的书有A，B，C，E。通过这些数据我们可以判断老张和老王的口味略相似，于是给老张推荐E这本书，同时给老王推荐D这本书

用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√

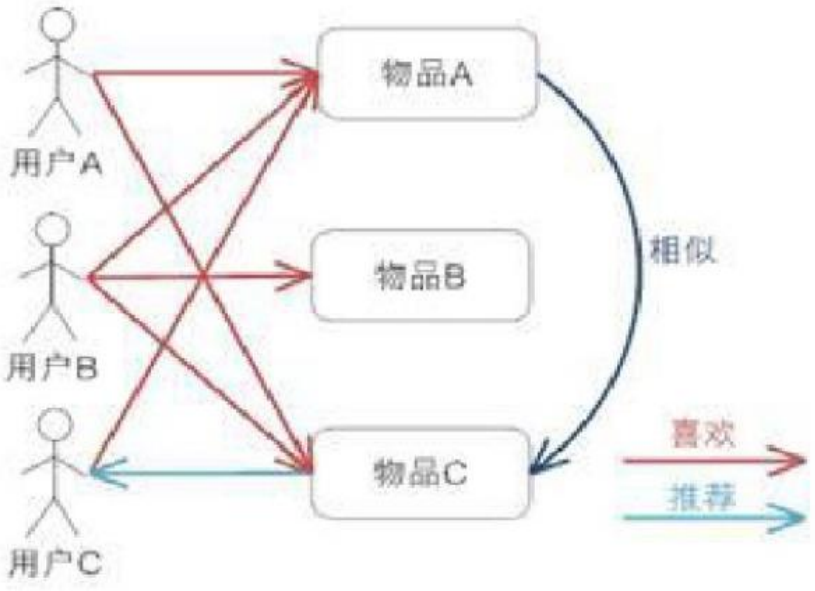
推荐系统——基于启发式的协同过滤

基于启发式的推荐算法（ collaborative filtering ）

基于项目的协同过滤

核心思想：基于用户对物品的偏好找到相似的物品，然后根据用户的历史偏好，为他推荐相似的物品

例如：我们发现喜欢看《从一到无穷大》的人大都喜欢看《什么是数学》，那么如果你刚津津有味地看完《从一到无穷大》，我们就可以立马给你推荐《什么是数学》



用户/物品	物品A	物品B	物品C
用户A	√		√
用户B	√	√	√
用户C	√		推荐

推荐系统——基于启发式的协同过滤

基于启发式的推荐算法（ collaborative filtering ）

优点

- **仅依赖用户的惯用数据，**需要很低程度的专业工程，能产生很好的效果
- 算法简单，高效

推荐系统——基于启发式的协同过滤

基于启发式的推荐算法（ collaborative filtering ）

缺点

- 倾向于推荐流行的物品，很难推荐给有独特口味的人，因为对某物品感兴趣的数据不够，该问题也被成为**流行性偏见**
- **冷启动问题**：用户评价商品数据少时，很难有效推荐；新商品评价的人少时，很难被有效推荐

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

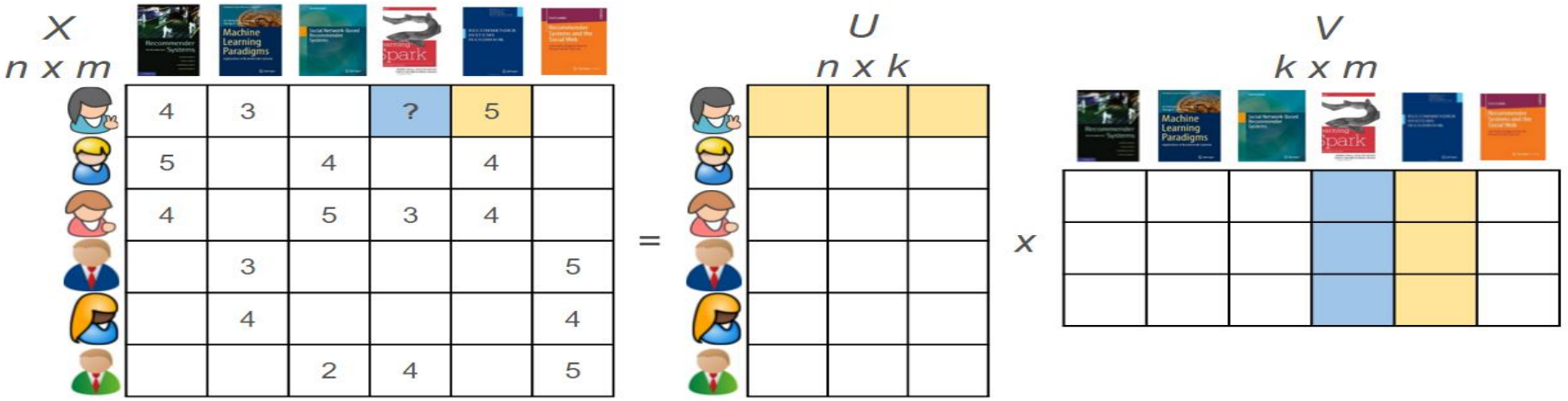
核心思想：将用户-物品矩阵分解为低秩的矩阵，可理解为抽取出潜在的影响因子，并通过这些因子描述用户和物品

- **SVD，SVD++**将评价矩阵分解为3个低秩的矩阵，这3个矩阵的乘积能对原始矩阵进行某种程度的复原，从而可以评估出缺失值
- **低秩矩阵**：在电影领域，这些自动识别的因子可能对应一部电影的常见标签，比如风格或者类型（戏剧片或者动作片），也可能是无法解释的

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

- 矩阵因子分解（如奇异值分解）将项和用户都转化成了相同的潜在空间，即将用户偏好矩阵分解成一个用户-潜在因子矩阵乘以一个潜在因子-项矩阵，代表用户和项之间的潜相互作用
- 矩阵分解背后的原理是潜在特征代表了用户如何给项进行评分。给定用户和项的潜在描述，我们可以预测用户将会给还未评价的项多少评分



推荐系统——基于模型的协同过滤

基于低秩矩阵分解

用户-潜在因子矩阵：表示不同的用户对于不用元素的偏好程度，1代表很喜欢，0代表不喜欢。如：

	小清新	重口味	优雅	伤感	五月天
张三	0.6	0.8	0.1	0.1	0.7
李四	0.1	0	0.9	0.1	0.2
王五	0.5	0.7	0.9	0.9	0

潜在因子-音乐矩阵：表示每种音乐含有各种元素的成分。如下表中，音乐A是一个偏小清新的音乐，含有小清新这个Latent Factor的成分是0.9，重口味的成分是0.1，优雅的成分是0.2……

	小清新	重口味	优雅	伤感	五月天
音乐A	0.9	0.1	0.2	0.4	0
音乐B	0.5	0.6	0.1	0.9	1
音乐C	0.1	0.2	0.5	0.1	0
音乐D	0	0.6	0.1	0.2	0

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

利用这两个矩阵，我们能得出张三对音乐A的喜欢程度是：

	小清新	重口味	优雅	伤感	五月天
张三	0.6	0.8	0.1	0.1	0.7

	小清新	重口味	优雅	伤感	五月天
音乐A	0.9	0.1	0.2	0.4	0

张三对小清新的偏好*音乐A含有小清新的成分
+对重口味的偏好*音乐A含有重口味的成分
+对优雅的偏好*音乐A含有优雅的成分+……

0.6*0.9+0.8*0.1+0.1*0.2+0.1*0.4+0.7*0=0.68

	音乐A	音乐B	音乐C	音乐D
张三	0.68	1.58	0.28	0.51
李四	0.31	0.43	0.47	0.11
王五	1.06	1.57	0.73	0.69

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

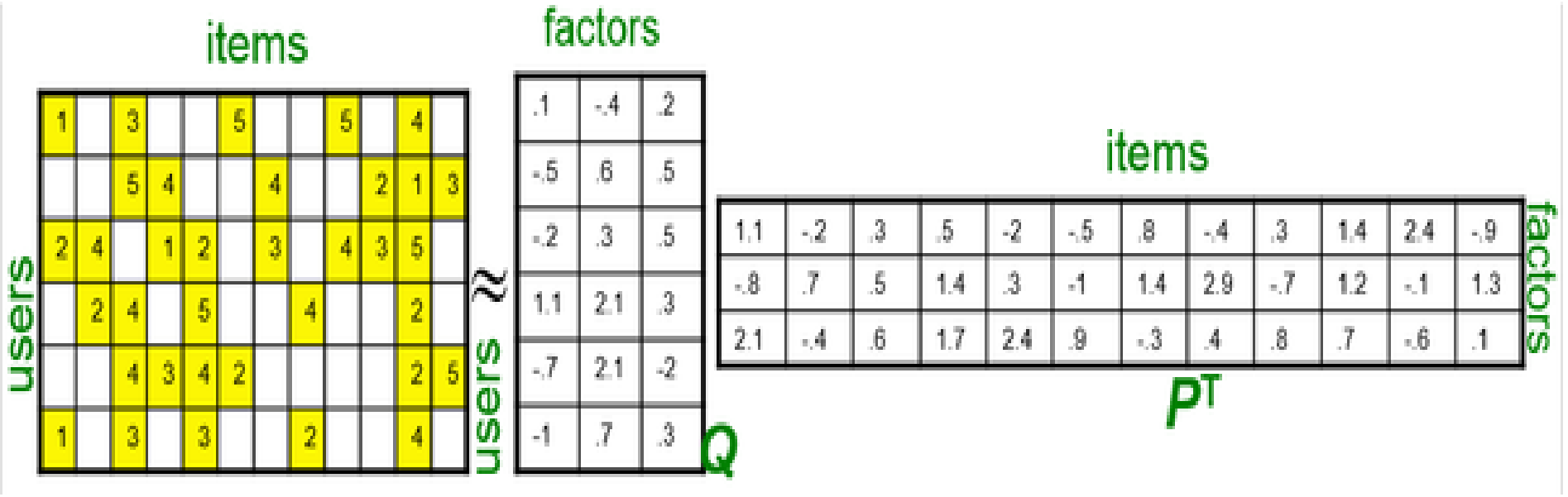
潜在因子（latent factor）是怎么得到的？

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

潜在因子（latent factor）是怎么得到的？



输入矩阵一般是非常稀疏的矩阵，通过奇异值分解等方法分解为低纬度的矩阵乘积

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

潜在因子（latent factor）是怎么得到的？

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

	因子1	因子2	因子3	因子4	因子5
用户1	0.908	0.642	0.524	0.454	0.406
用户2	0.877	0.620	0.506	0.438	0.392
用户3	0.768	0.543	0.443	0.384	0.344
用户4	0.853	0.603	0.492	0.426	0.381
用户5	0.847	0.599	0.489	0.424	0.379
用户6	0.884	0.625	0.510	0.442	0.395
用户7	0.870	0.615	0.502	0.435	0.389
用户8	0.878	0.621	0.507	0.439	0.392

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
	0.914	0.913	0.906	0.921	0.850	0.900	0.919	0.937	0.931	0.947	0.891	0.937	0.900
	0.646	0.645	0.640	0.652	0.601	0.636	0.650	0.663	0.658	0.670	0.630	0.663	0.636
	0.528	0.527	0.523	0.532	0.491	0.520	0.531	0.541	0.537	0.547	0.514	0.541	0.520
	0.457	0.456	0.453	0.461	0.425	0.450	0.460	0.469	0.465	0.473	0.445	0.469	0.450
	0.409	0.408	0.405	0.412	0.380	0.402	0.411	0.419	0.416	0.423	0.398	0.419	0.402

推荐系统——基于模型的协同过滤

基于低秩矩阵分解

估计的得分矩阵

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1		2.10	2.08	2.12	1.96		2.12	2.16			2.05		
用户2	2.03	2.03	2.01		1.89	2.00	2.04	2.08		2.10	1.98	2.08	
用户3	1.78	1.78		1.80				1.83	1.82			1.83	
用户4	1.98				1.84	1.95		2.03		2.05	1.93		1.95
用户5	1.96			1.98		1.93			2.00	2.04		2.01	1.93
用户6	2.05	2.04		2.06	1.90		2.06	2.10		2.12	2.00	2.10	2.02
用户7	2.02		2.00	2.03	1.87		2.08	2.07	2.05		1.96		
用户8	2.03		2.01	2.05	1.89		2.04		2.07		1.98	2.09	

实际评分矩阵R

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

推荐系统——基于模型的协同过滤

基于低秩矩阵分解的推荐算法

优点

- 不需要对物品或者用户进行严格建模，而且不要求物品的描述是机器可理解的，所以这种方法也是领域无关的
- 很好的支持用户发现潜在的兴趣偏好

推荐系统——基于模型的协同过滤

基于低秩矩阵分解的推荐算法

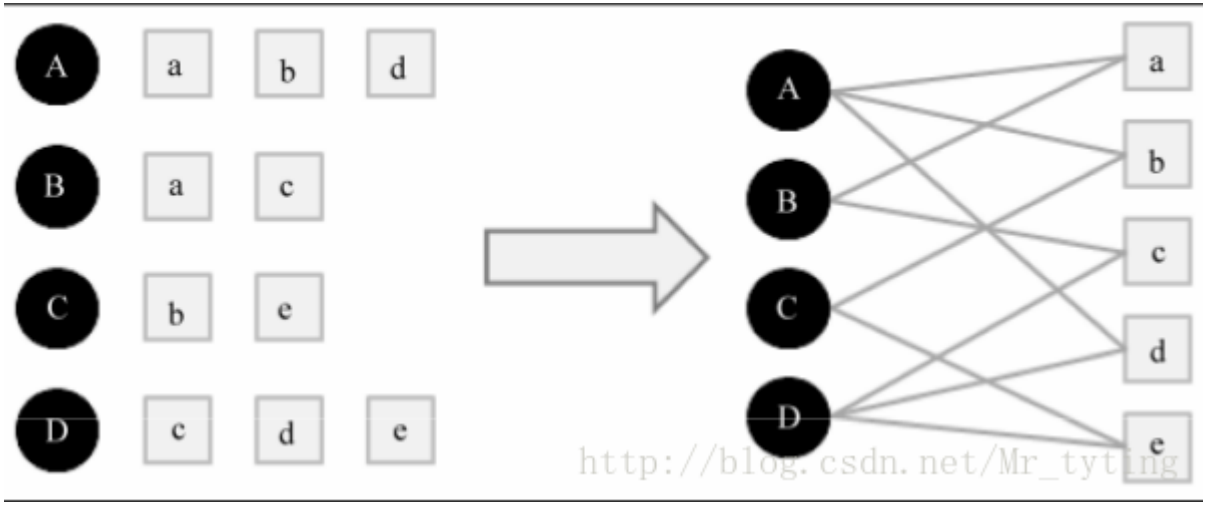
缺点

- 方法的核心是基于历史数据，所以对新物品和新用户都有“冷启动”的问题
- 推荐的效果依赖于用户历史偏好数据的多少和准确性
- 在大部分的实现中，用户历史偏好是用稀疏矩阵进行存储的，而稀疏矩阵上的计算有些明显的问题，包括可能少部分人的错误偏好会对推荐的准确度有很大的影响等
- 对于一些特殊品味的用户不能给予很好的推荐
- 由于以历史数据为基础，抓取和建模用户的偏好后，很难修改或根据用户的使用演变，从而导致这个方法不够灵活

推荐系统——基于图的推荐算法

将用户行为数据用二分图表示，例如用户数据是由一系列的二元组组成，其中每个元组(u,i)表示用户u对物品i产生过行为

- 给用户u推荐物品任务可以转化为度量节点Uv和与Uv 没有边直接相连 的物品节点在图上的相关度，**相关度越高的在推荐列表中越靠前**
- 两个顶点的相关度主要取决于如下因素：
  - 两个顶点之间路径数
  - 两个顶点之间路径长度
  - 两个顶点之间路径经过的顶点



推荐系统——混合推荐方法

- 由于各种推荐方法都有其自身的优缺点，因此在实际中经常使用混合推荐（Hybrid Recommendation）的方法。混合推荐的一个最重要原则就是通过组合后应能避免或弥补各自推荐技术的弱点

- 研究和应用最多的是内容推荐和协同过滤推荐的组合。最简单的做法就是分别用基于内容的方法和协同过滤推荐方法去产生一个推荐预测结果，然后用某方法组合其结果

- 尽管从理论上有很多种推荐组合方法，但在某一具体问题中并不见得都有效，不同的组合思路适用于不同的应用场景

推荐系统——基于关联规则的推荐

- 基于关联规则的推荐（Association Rule-based Recommendation）是以关联规则为基础，把已购商品作为规则头,规则体为推荐对象

- 关联规则就是在一个交易数据库中统计购买了商品集X的交易中有多大比例的交易同时购买了商品集Y，其直观的意义就是用户在购买某些商品的时候有多大倾向去购买另外一些商品。比如购买牛奶的同时很多人也会购买面包。关联规则“X->Y”表示对于购买了商品X的用户，系统将给他推荐商品Y

- 关联规则挖掘可以发现不同商品在销售过程中的相关性，在零售业中已经得到了成功的应用

- 关联规则的发现是算法的第一步也是最为关键且最耗时的，是算法的瓶颈，但可以离线进行。其次，商品名称的同义性问题也是关联规则的一个难点

推荐系统——基于效用的推荐

- 基于效用的推荐（Utility-based Recommendation）是建立在对用户使用项目的效用情况上计算的，其核心问题是怎么样为每一个用户去创建一个效用函数

- 用户资料模型很大程度上是由系统所采用的效用函数决定的

- 基于效用推荐的好处是它能把非产品的属性，如提供商的可靠性（Vendor Reliability）和产品的可得性（Product Availability）等考虑到效用计算中

推荐系统——基于知识的推荐

□基于知识的推荐（Knowledge-based Recommendation）在某种程度是可以看成是一种推理（Inference）技术，它不是建立在用户需要和偏好基础上进行推荐的。而是利用针对特定领域制定规则(rule)来进行基于规则和实例的推理(case-based reasoning)

□效用知识 (functional knowledge) 是一种关于一个对象如何满足某一特定用户的知识, 能够解释需求和推荐的关系，用于推荐系统。效用知识在推荐系统中必须以机器可读的方式存在(ontology本体知识库)

•文献[1]中利用饭店的菜式方面的效用知识，推荐饭店给顾客；

[1]Burke R. Knowledge-Based recommender systems. Encyclopedia of Library and Information Systems, 2000, 69(32): 180–200.

• 文献[2]使用关于学术论文主题的ontology本体知识库向读者作推荐

[2]Middleton SE, Shadbolt NR, de Roure DC. Ontological user profiling in recommender systems. ACM Trans. on Information Systems, 2004,22(1):54–88

推荐系统——推荐算法总结

推荐方法	优点	缺点
基于内容推荐	推荐结果直观，容易解释； 要求特征内容有良好的结构性。	新用户问题； 复杂属性不好处理； 要有足够数据构造分类器
协同过滤推荐	新异兴趣发现、不需要领域知识； 随着时间推移性能提高； 推荐个性化、自动化程度高； 能处理复杂的非结构化对象	稀疏问题； 可扩展性问题； 新用户问题； 质量取决于历史数据集； 系统开始时推荐质量差；
基于规则推荐	能发现新兴趣点； 不要领域知识	规则抽取难、耗时； 产品名同义性问题； 个性化程度低；
基于效用推荐	无冷启动和稀疏问题； 对用户偏好变化敏感； 能考虑非产品特性	用户必须输入效用函数； 推荐是静态的，灵活性差； 属性重叠问题；
基于知识推荐	能把用户需求映射到产品上； 能考虑非产品属性	知识难获得； 推荐是静态的

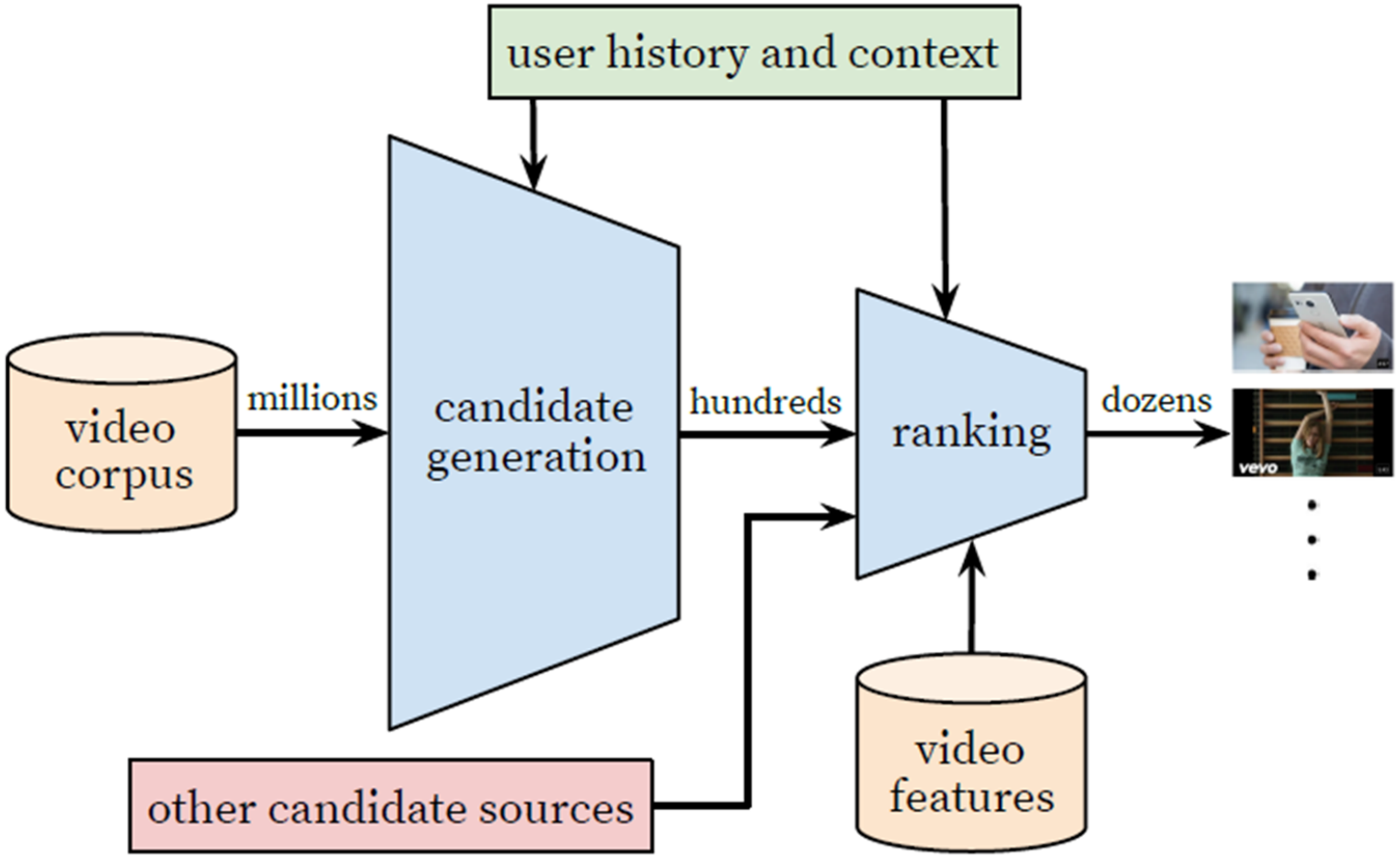
推荐系统——其他算法

- 深度学习
- 社会化推荐
- 学习排序
- 多臂赌博机（Multi-Bandit）
- 张量因子分解
- ……

提纲

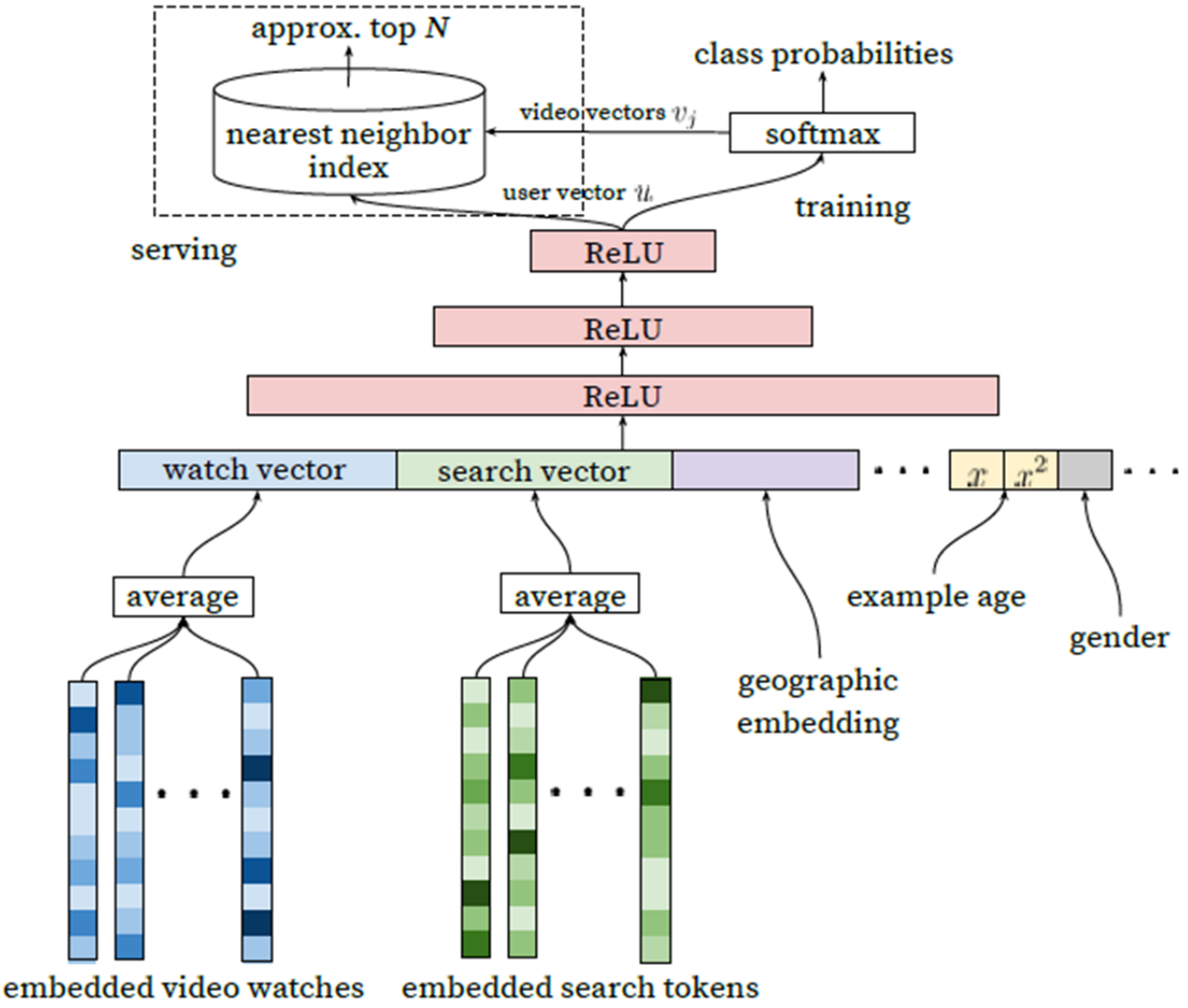
1. 推荐系统
2. Youtube视频推荐

Youtube-深度学习架构



Youtube-深度学习架构

候选集生成



Youtube-深度学习架构

视频名编码

Q:关于编码

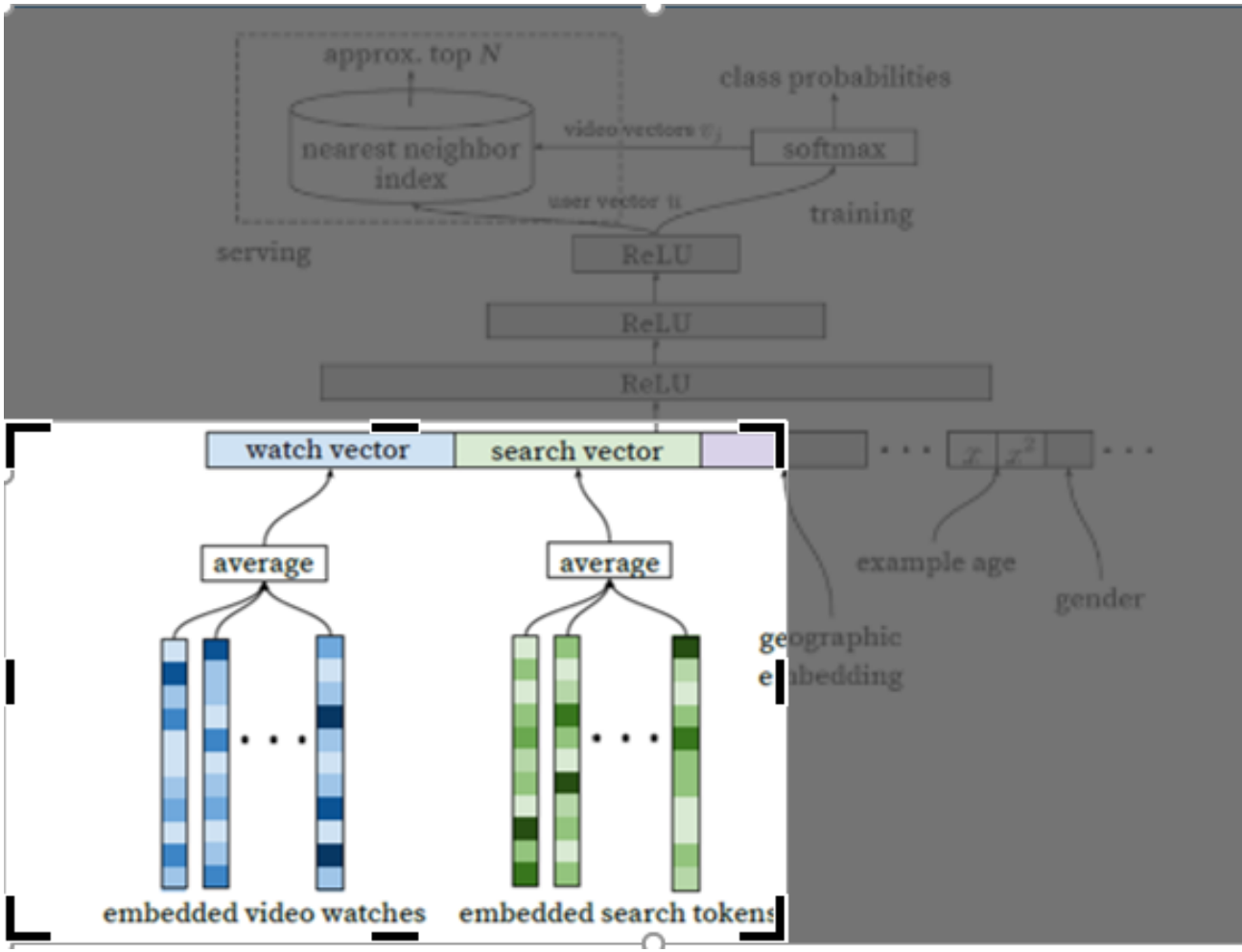
A:不是0-1稀疏编码，是采用word2vec编码方式，从视频ID词汇中计算出一个固定长度的多维向量,用来描述该视频

Q:关于平均（Average）

A:不是直接将向量取平均值，而是根据时间和权重计算

Q:输入视频是根据所有视频编码吗？

A:不是，只包含用户看过的视频以及搜索记录ID，不是真实视频帧



Youtube-深度学习架构

用户&视频信息

Q:geographics embedding
A:geographics指地理信息特征，比如你的定位信息；而Demographic指的是人口统计学信息，因此gender与geographic不冲突

Q:example age (Average)
A:根据之前视频编码的权重以及视频上传后的时间去计算

$$age = \sum_{i}^{n} \alpha * age_i$$

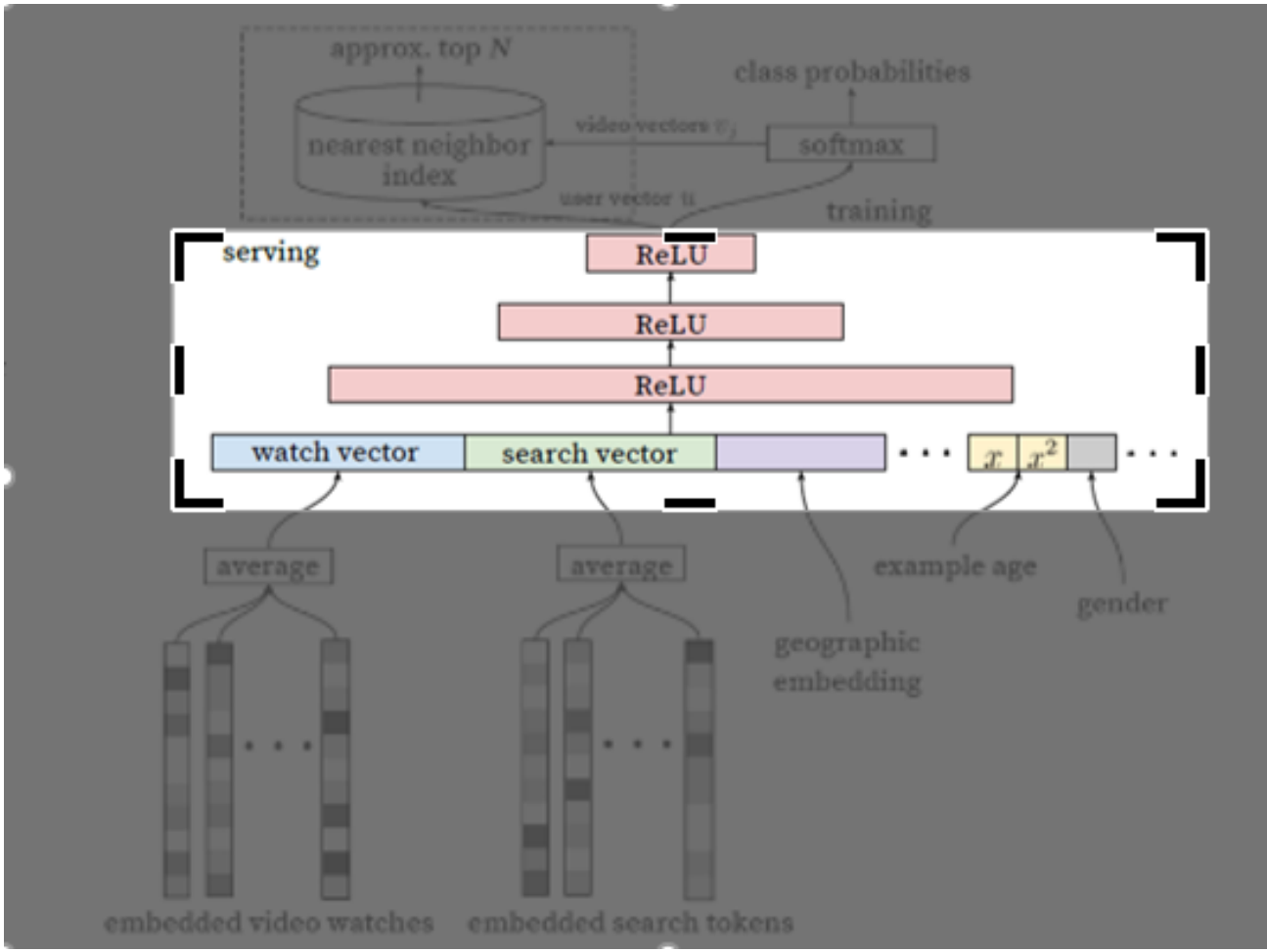
Youtube-深度学习架构

全连接层

Q:Relu

A:避免过拟合的一种方式

$$f(x) = \frac{1}{1+exp(-x)}$$



Youtube-深度学习架构

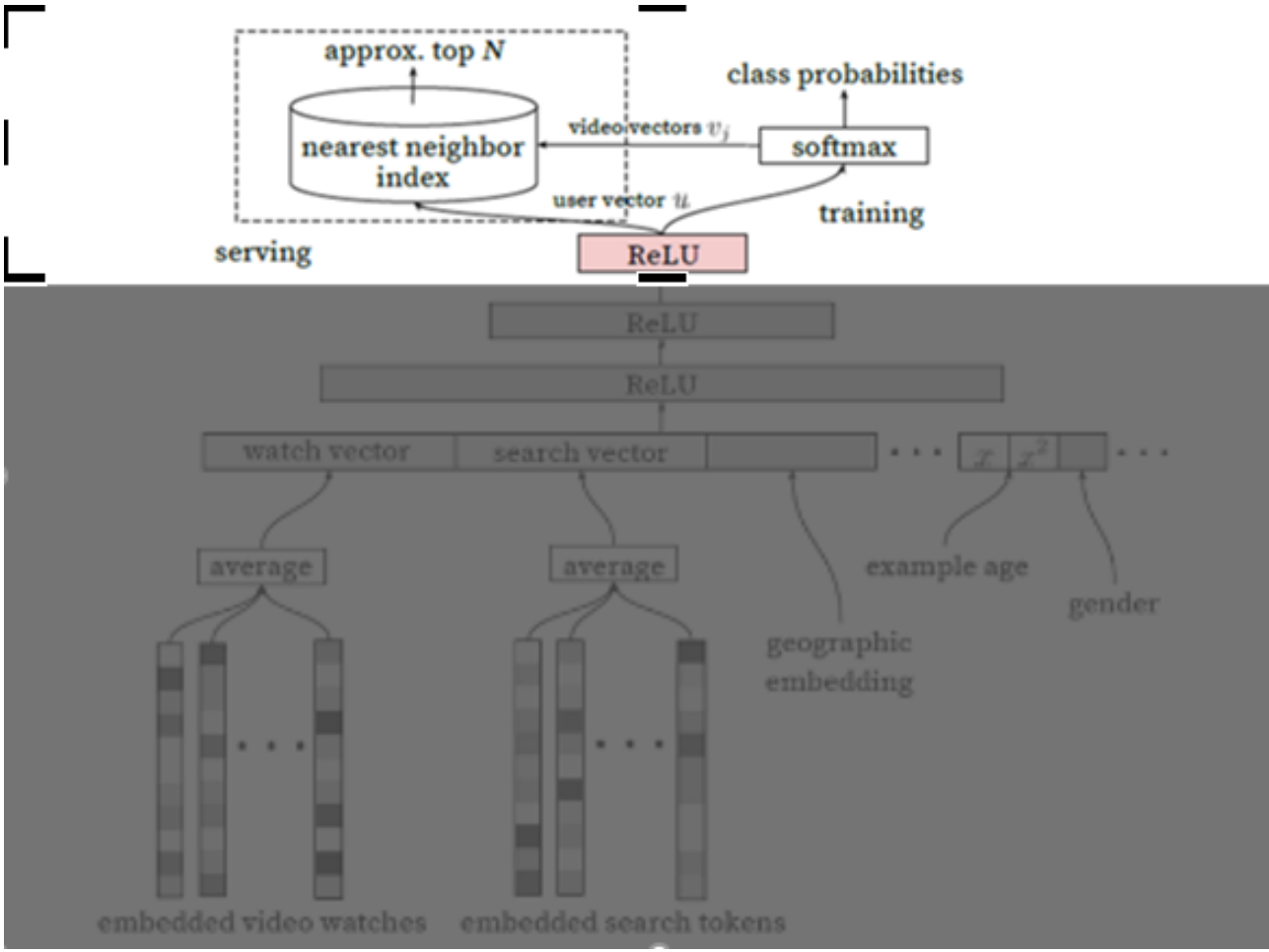
全连接层

Q:最终产生的是什么？

A:视频向量Vj,用户向量Uj,然后根据协同过滤从海量视频库中选出所需要的视频。

Q:视频的类别是什么 ？

A:神经网络训练的的出的视频向量，没有一个固定的类别，由于向量之间具有相似性，因此可以通过特征之间的距离或者cos值进行度量。



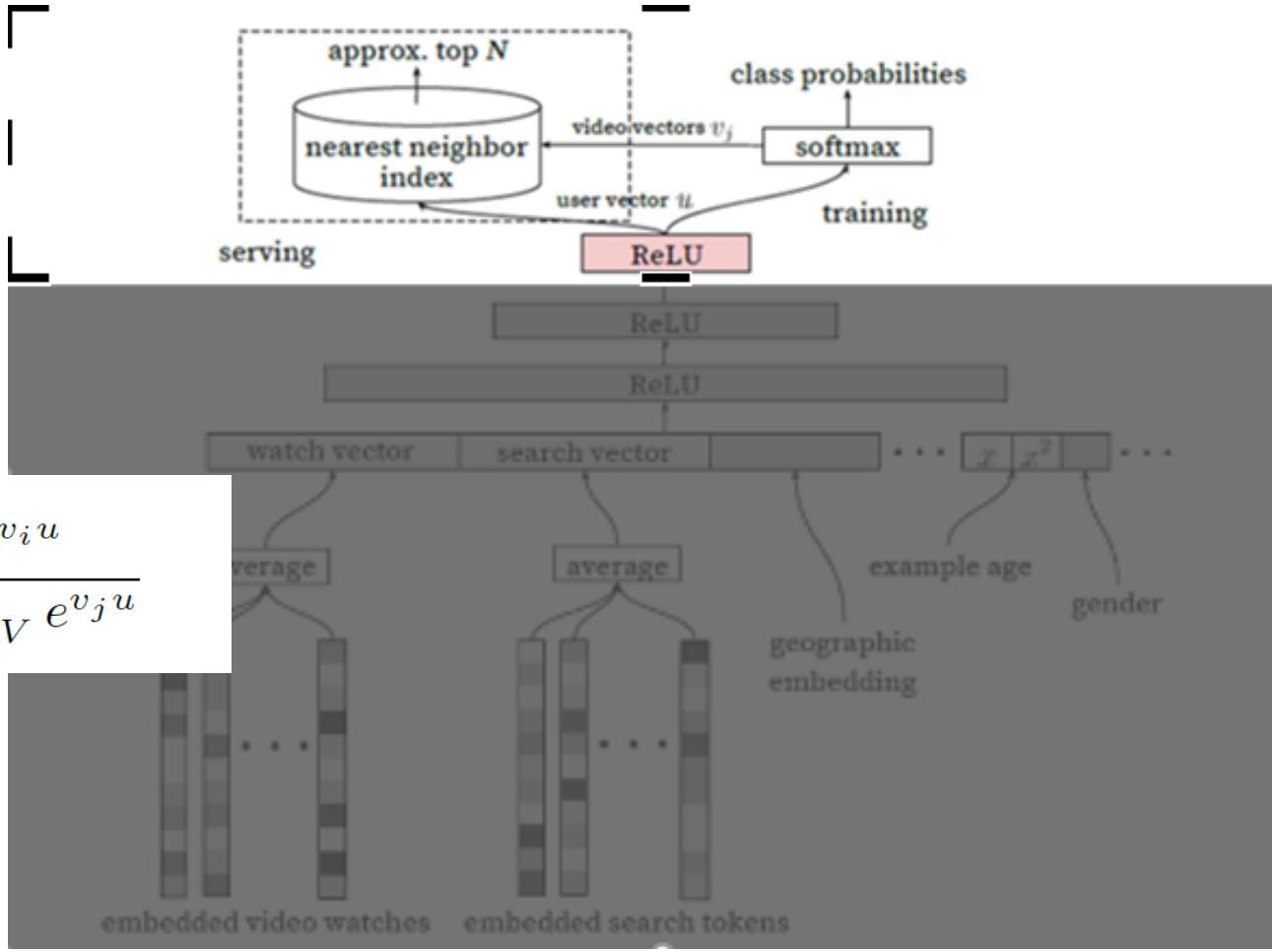
Youtube-深度学习架构

全连接层

Q:分类概率是如何计算的 ？

A:基于特定的用户U和上下文C，在时间t将指定的视频wt准确地划分到第i类中，其中i∈V

$$P(w_t = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

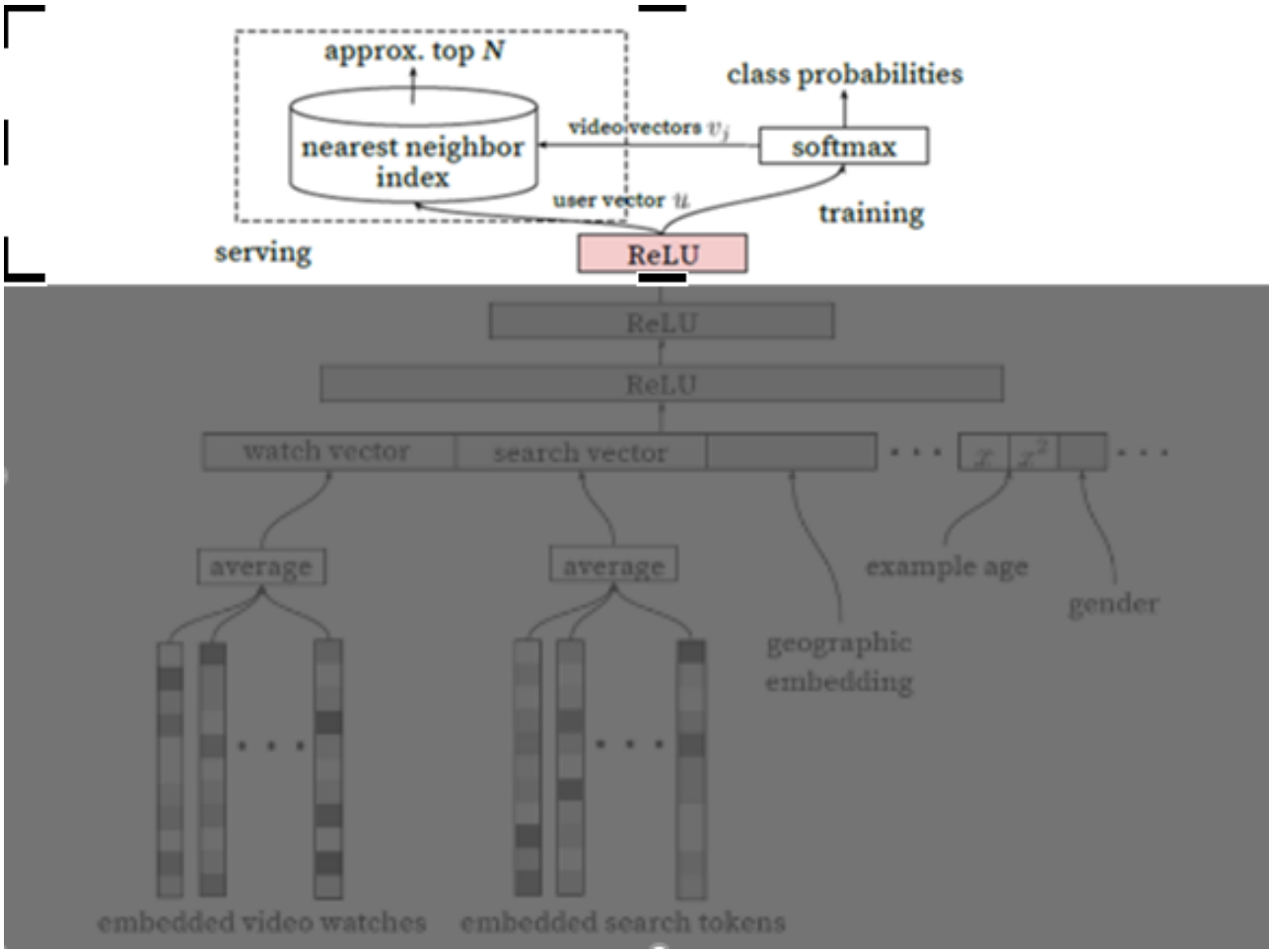


Youtube-深度学习架构

全连接层

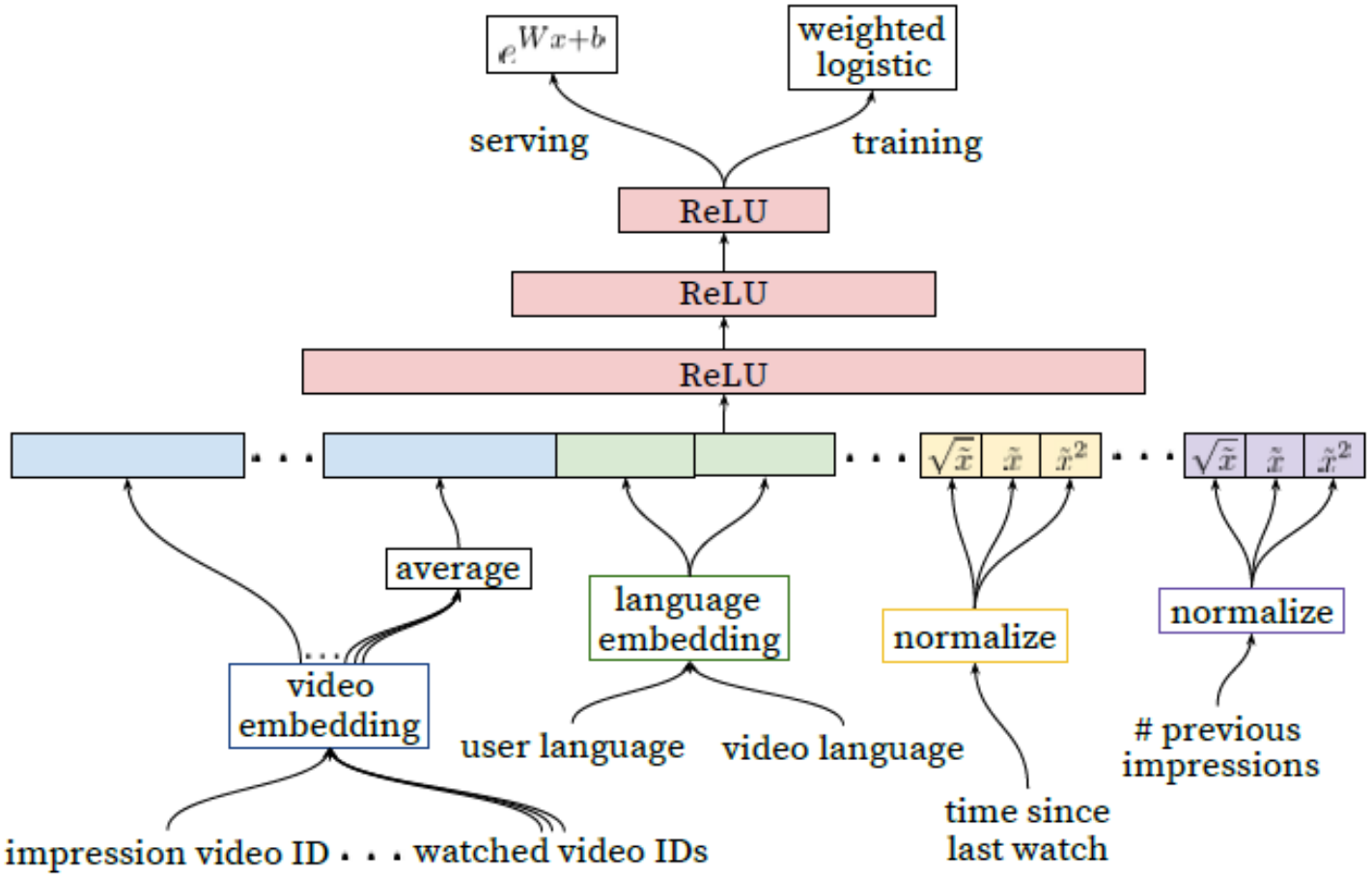
Q:网络模型是如何训练的 ？

A:根据用户已知的历史行为，比如前t-1时刻用户的观看记录以及搜索记录，通过网络计算出第t时刻的推荐类别max（t）,与用户真实的观看记录所产生的视频向量t进行比较。反向传播进行训练。



Youtube-深度学习架构

排序模型

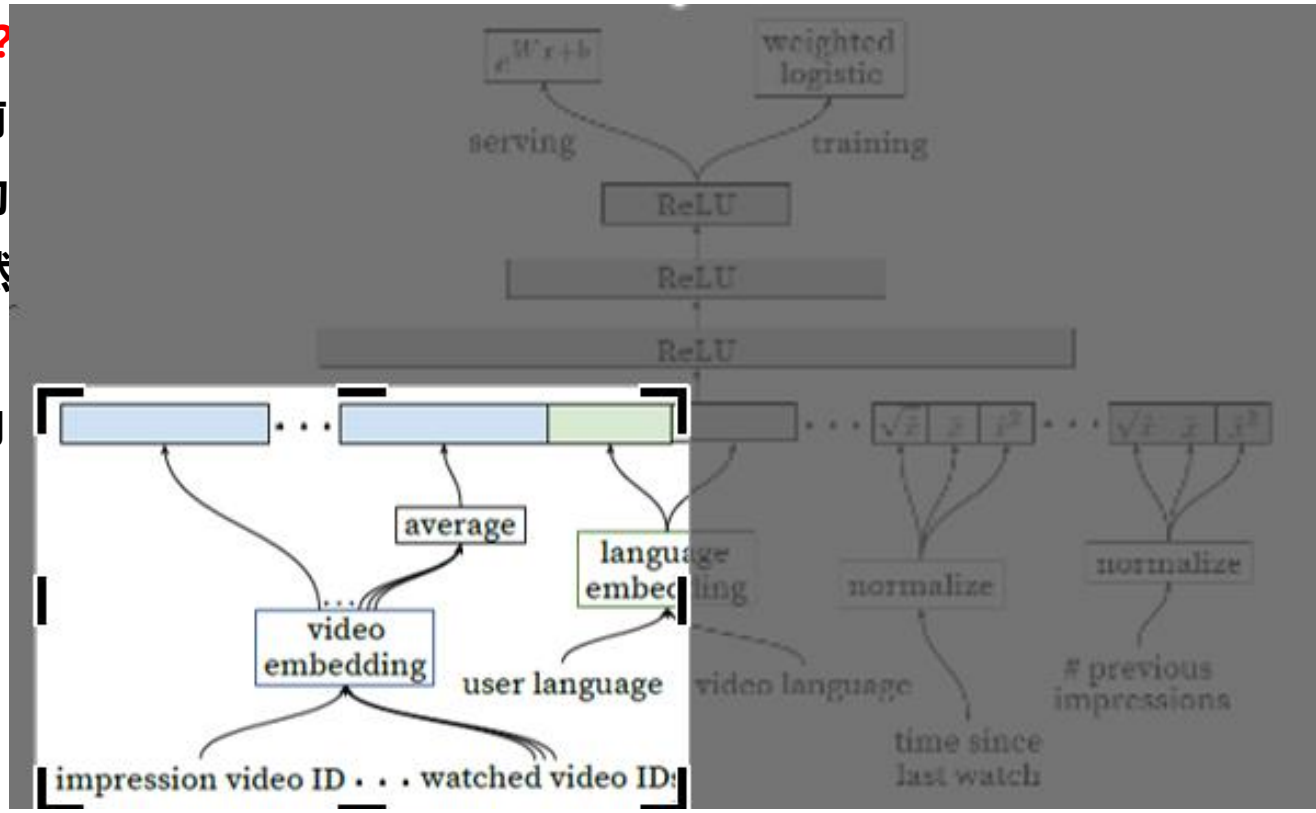


Youtube-深度学习架构

输入特征

Q:视频向量如何作为输入的 ？

A:根据上一步候选集得到的前n个视频 ID，以及用户观看的视频ID做一次embedding,然后取出其中定长的k个视频 向量以及加权平均之后的视频向量联合作为视频ID

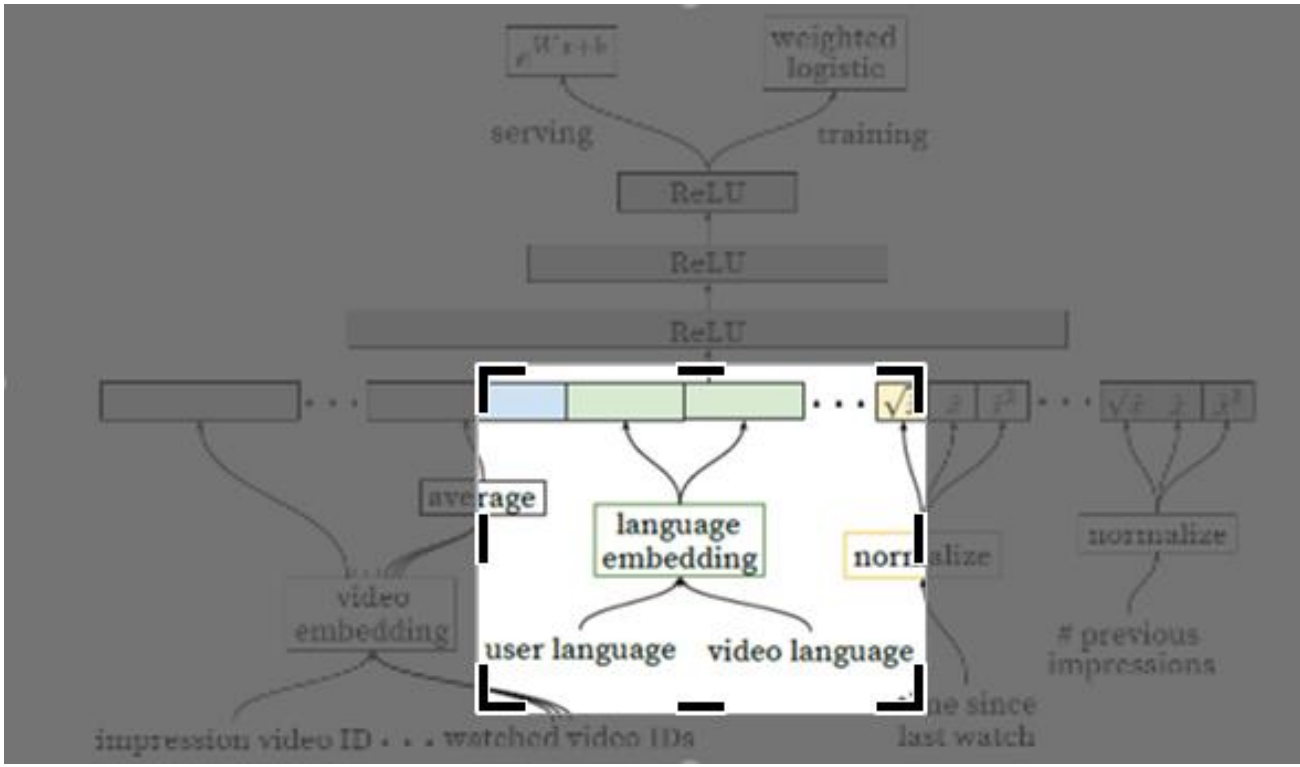


Youtube-深度学习架构

输入特征

Q:language embedding是指什么 ？

A:视频包括用户自己的语言，比如中文，以及推荐的视频自身的语言比如英文等等，两者embedding之后结合作为输入特征



Youtube-深度学习架构

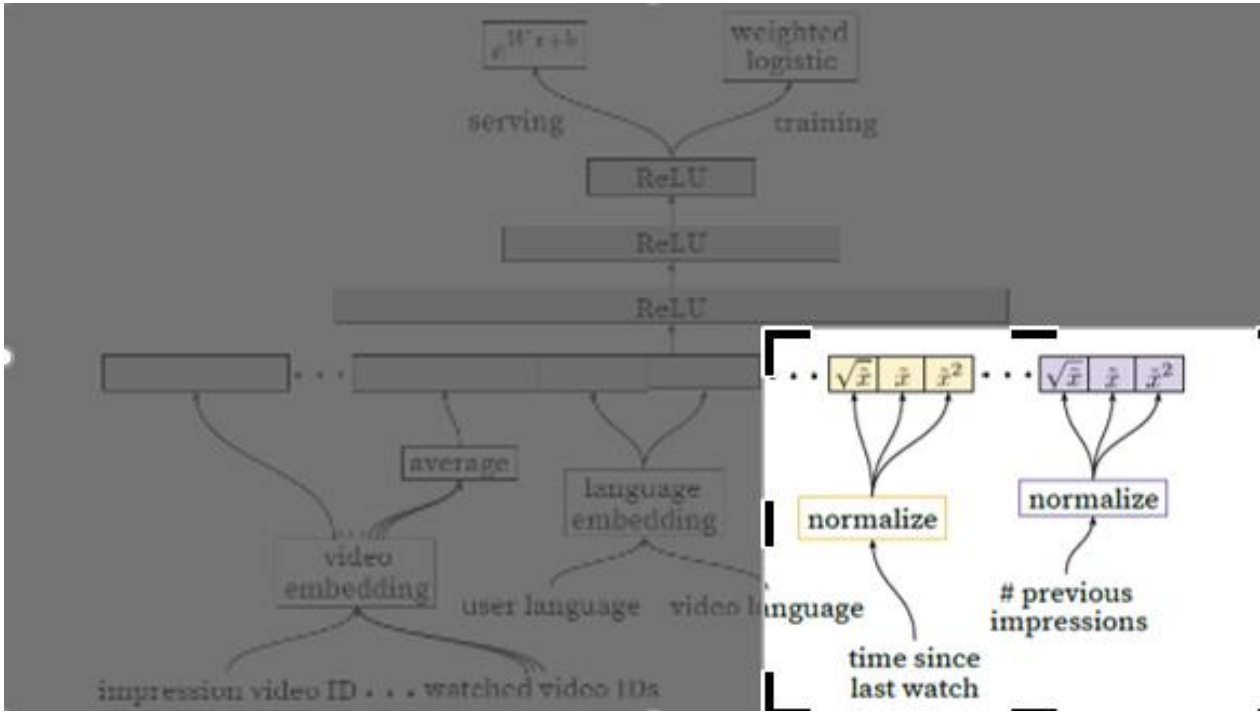
输入特征

Q:用户上次观看的视频时间以及之前推荐没有看的视频数目正则化是指什么 ？

A:正则化是为了把这些特征值转换成0-1之间的值，具体方式是通过概率 分布去计算

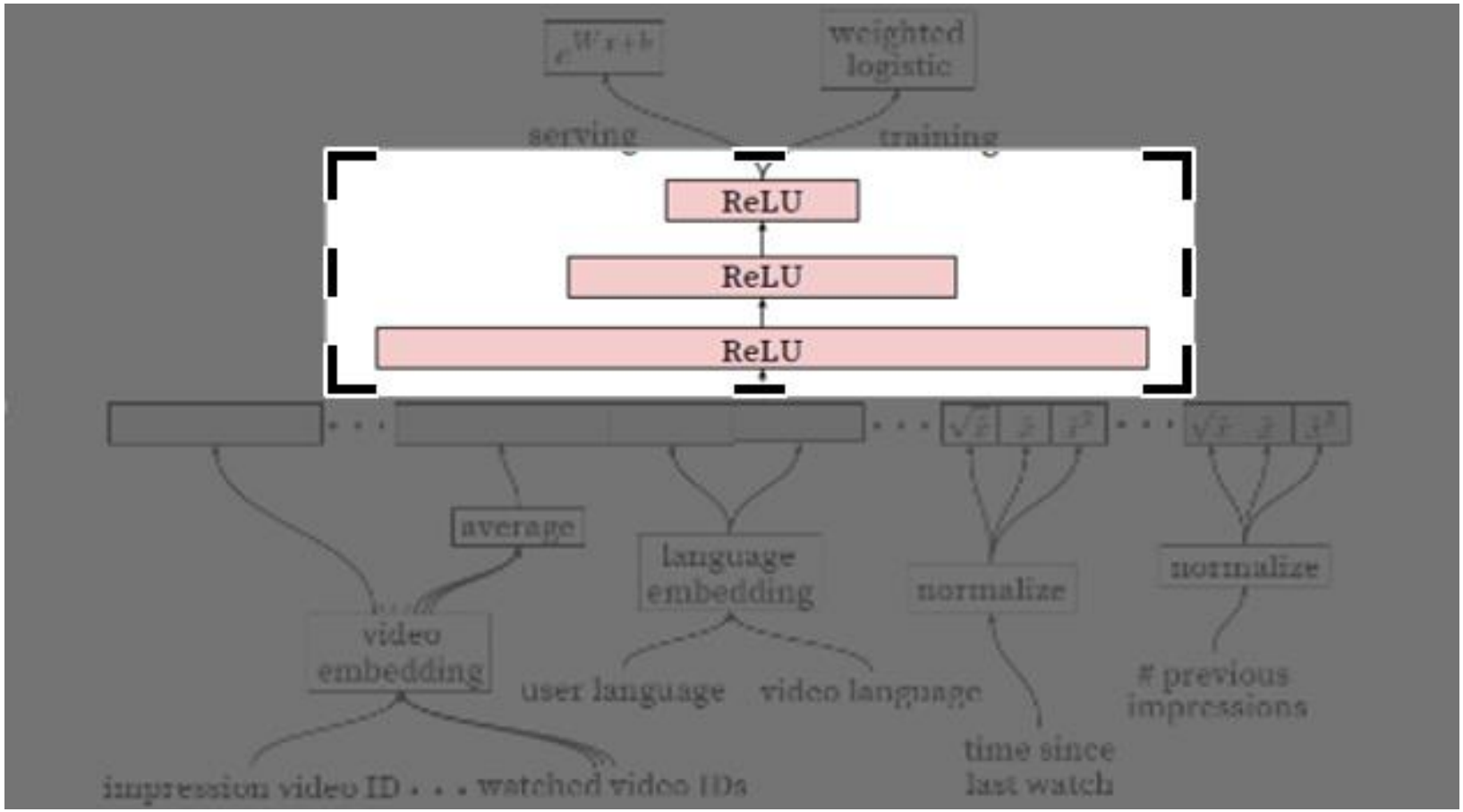
Q:为什么要开根号和取平方值 ？

A:实验过程中防止特征权重太少 或者线性原因导致对最终的视频 训练效果 影响太小，所以对其进行非线性处理并且把向量结合作为输入



Youtube-深度学习架构

网络训练

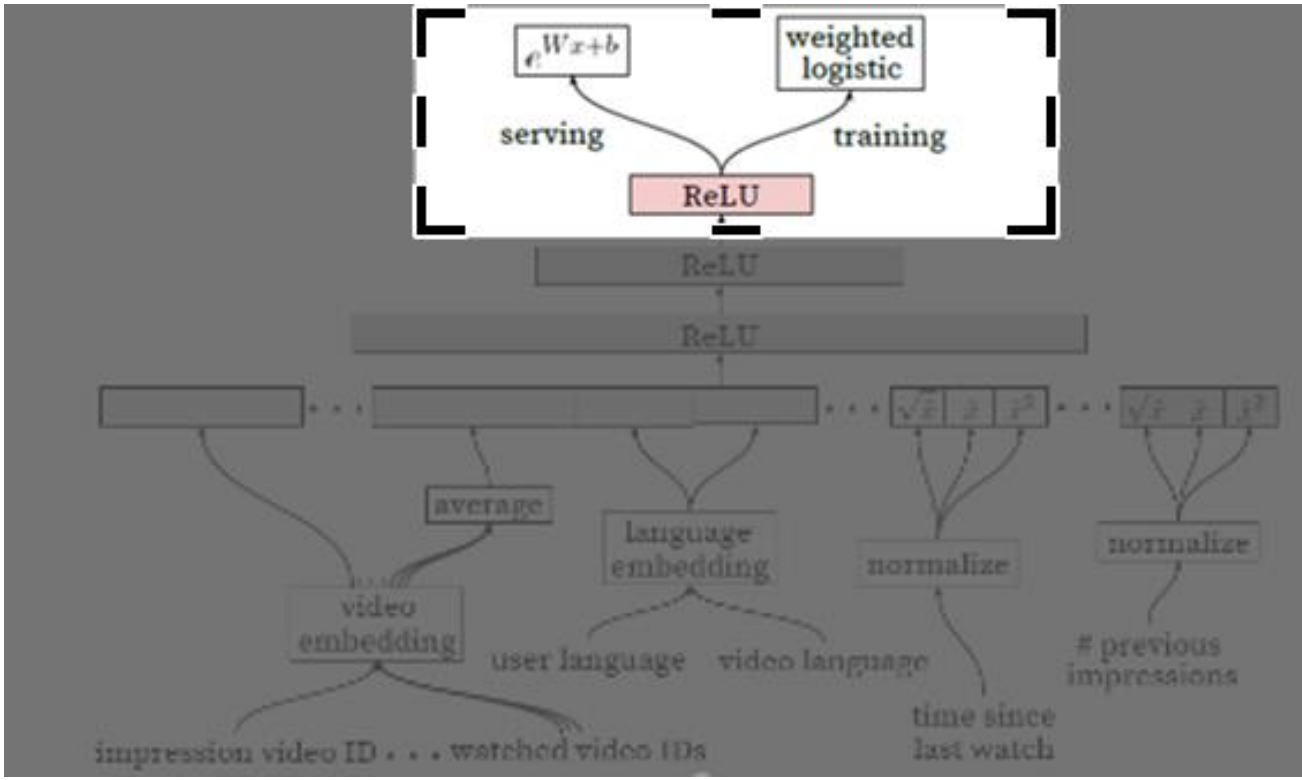


Youtube-深度学习架构

网络训练

Q:网络结构如何初始化的？

A:通过网络期望得到的视频观看时间与用户实际观看视频时间进行对比，产生一个初始的网络权重，跟之前候选集的生成初始化过程类似。

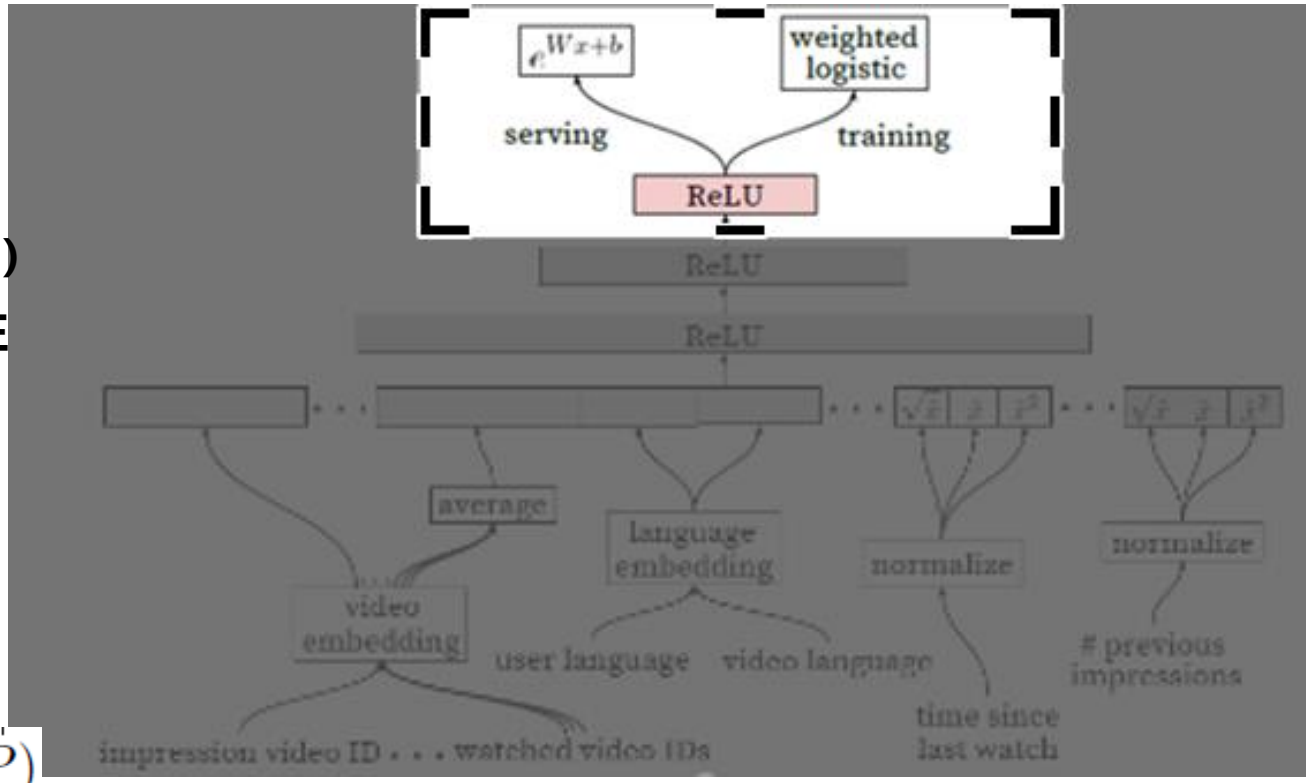


Youtube-深度学习架构

网络训练

Q:网络结构如何训练的？

A:通过A/B测试以及用户对推荐视频的行为进行反馈，这里主要是计算正负样本（共N个）的时间来表示，更加 准确，正样本（k个）是推荐给用户用户看了的视频，复样本是用户没有观看的视频，计算方式如下：



$$\frac{\sum T_i}{N-k} \qquad \mathrm{E}[T](1+P)$$

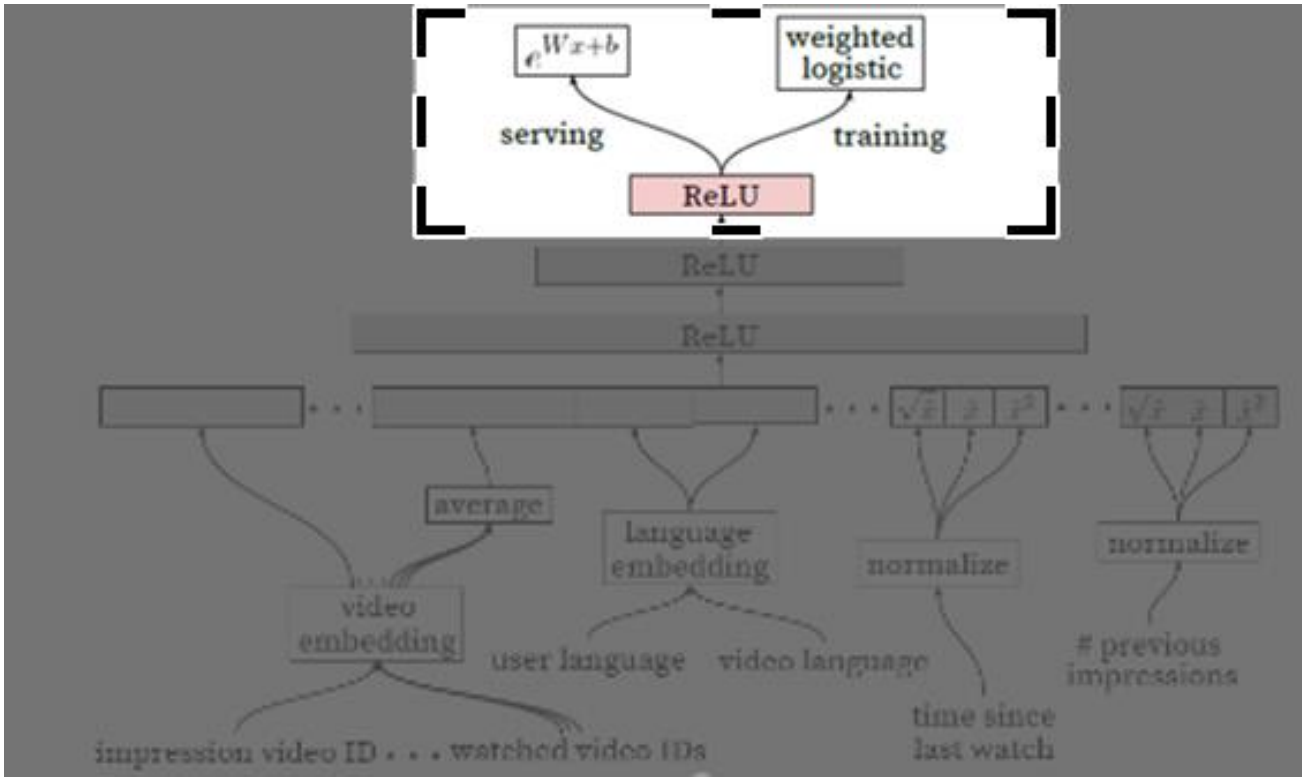
N表示的是训练样本的数目，k表示的是正样本的数目，Ti表示的是第i个展示被观看的时长。

Youtube-深度学习架构

网络训练

Q:网络最终的输出是什么？

A:最终输出的训练得到的视频向量的权重，并通过逻辑回归对权重进行排序，按照排序大小对视频进行推荐。



END