

第三章 基于多信号融合的分解推荐模型

3.1 问题定义

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。

在本章，我们提出一种基于聚类信息融合的推荐模型，它属于基于模型的协同过滤推荐算法。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。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

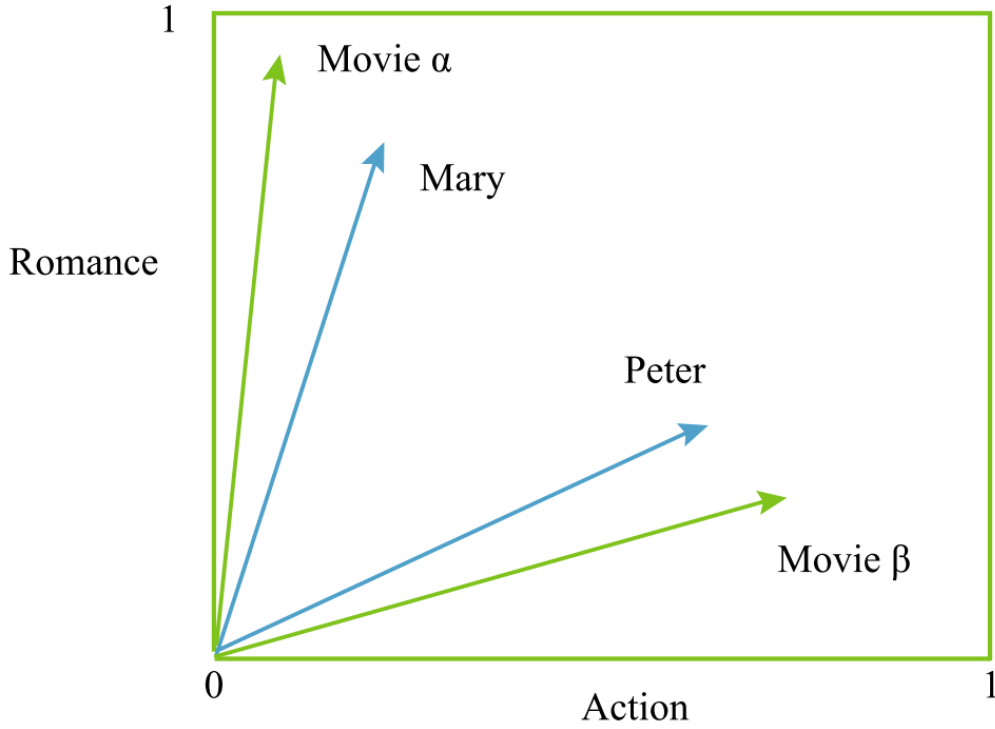


Figure 3.1: 依据隐含变量进行选择的过程

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 α ，而更喜欢电影 β ，对于Mary来说正好相反。一旦向量被计算好，推荐就能够很轻松地进行。如果我们使用 K 个隐含因子，那么隐含特征响亮就是 K 维，那么如果 $K(N + M) < NM$ ，那么相关的数据被大量的减少，从而达到维度下降的效果。

在实际情况中，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。

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。者需要保证两点，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。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。3.3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。

3.2 算法简介

3.2.1 动机与直觉

我们的算法是在BPR-MF(基于矩阵分解的贝叶斯排序推荐)之上，结合视频聚类信息的多信号源推荐系统。之所以是“多信号”，在于原来的BPR-MF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

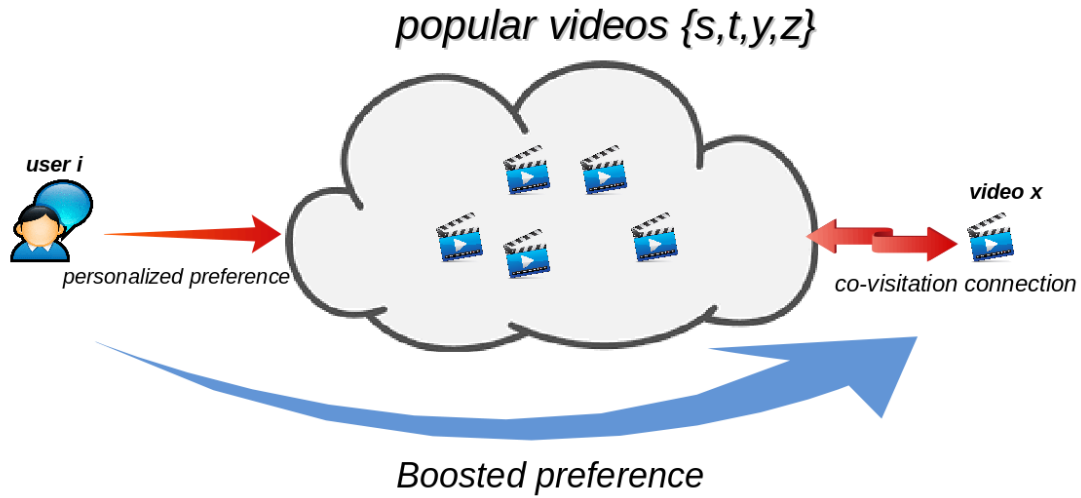


Figure 3.2: clustering boosted preference

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。

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3.2所示，视频 x 与视频 s,t,y,z 紧紧地关联在一起(每次 x 的出现都伴随着视频 s,t,y,z 的出现),但是相对于所有的数据， x 的训练数据相对较少。假设我们要对某个观看过视频 s,t,y,z 的用户做推荐，因为视频 x 的高度相关性，我们希望 x 会被推荐.如果仅提供偏序信号，那么 x 很难被推荐到，因为其他流行的但却相关性不强的视频会将 x 推送到底部。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

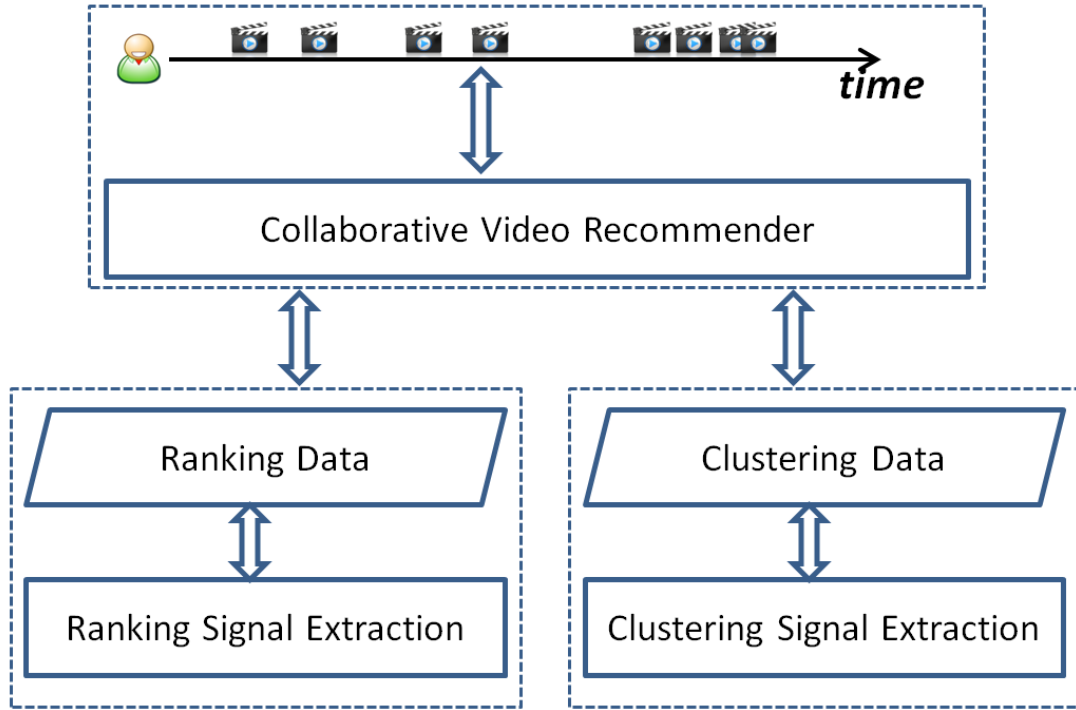


Figure 3.3: overview of model

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：

$$mix_loss = ranking_loss + clustering_loss \quad (3.1)$$

如图3.3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3.5节将会介绍整合模型的细节。

3.3 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。

3.3.1 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:

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

为了方便起见,做如下定义:

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

3.3.2 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 $(U \times I \setminus S)$ 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 $(U \times I \setminus S)$ 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3.4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：

$$D_S := \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\} \quad (3.4)$$

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。

3.3.3 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。对于用户而言，它们就对应着用户对喜剧的喜爱程度以及对动作武打片的喜爱程度。相对应的评分函数如下：

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

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。

3.3.4 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：

$$p(\Theta | >_u) \propto p(>_u | \Theta) p(\Theta) \quad (3.6)$$

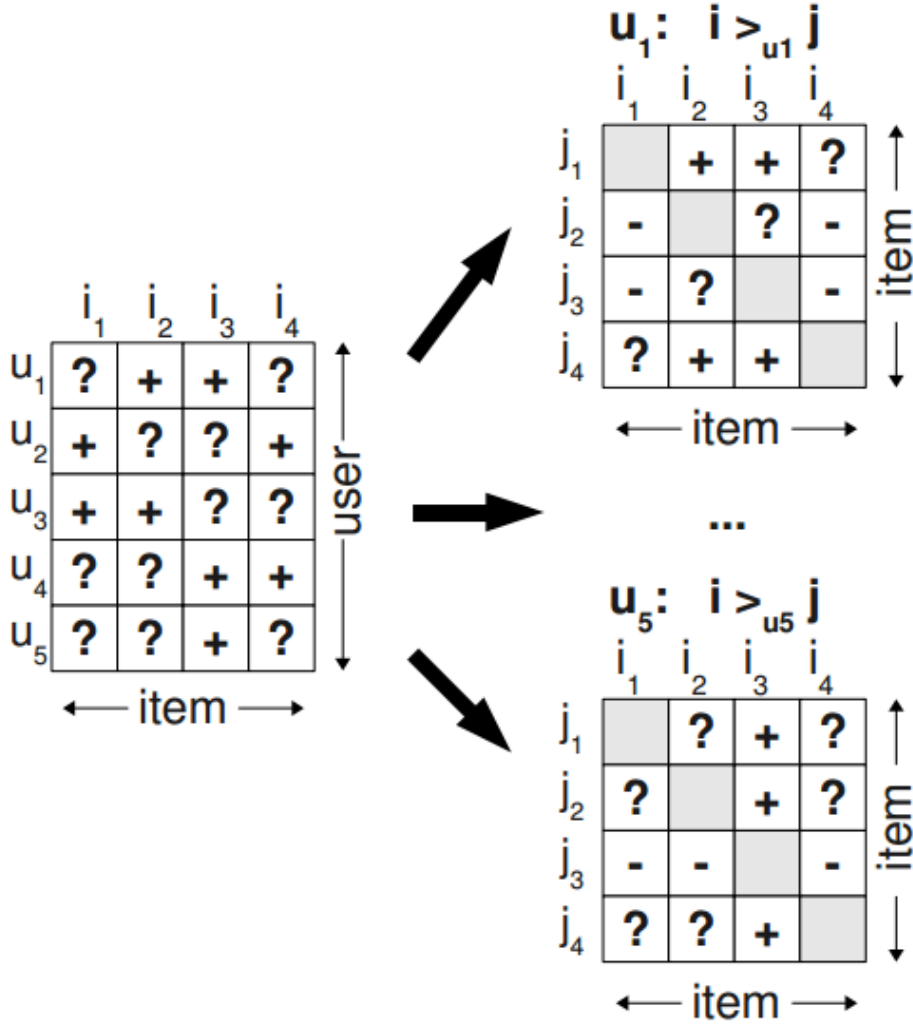


Figure 3.4: data processing of BPR

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：

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

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：

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

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

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

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。

3.4 聚类的引入

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

3.4.1 聚类模型

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



Figure 3.5: pairwise cluster extraction

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：

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

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。

3.4.2 聚类信号的截取

如之前所言，我们将视“共视”行为作为聚类反馈。在视频网站里，用户的观看行为都会有对应的时间戳对应。对每个用户，我们将视频在时间轴上排序，在时间轴上维护一个时间区间为 δt 的移动时间窗口。对于在时间窗内的任意两个视频 $\langle j, k \rangle$ ，我们取出三元组 $\langle i, j, k \rangle$ 作为一个聚类信号，如3.5所示。

δt 的取值决定了对聚类的阈值以及所能获取的聚类信号的数量。我们不能将 δ 设置得太大，如24h。因为太大的时间窗会将多次会话中的观看行为归入一

次聚类中，造成聚类信号中混入大量噪音；我们同样不能将其值设置的太小，因为那样将得不到足够的聚类信号。

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

3.4.3 聚类信号的优化

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

受[2]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：

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

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：

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

3.5 整合模型

3.5.1 信号的整合

图3.6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。

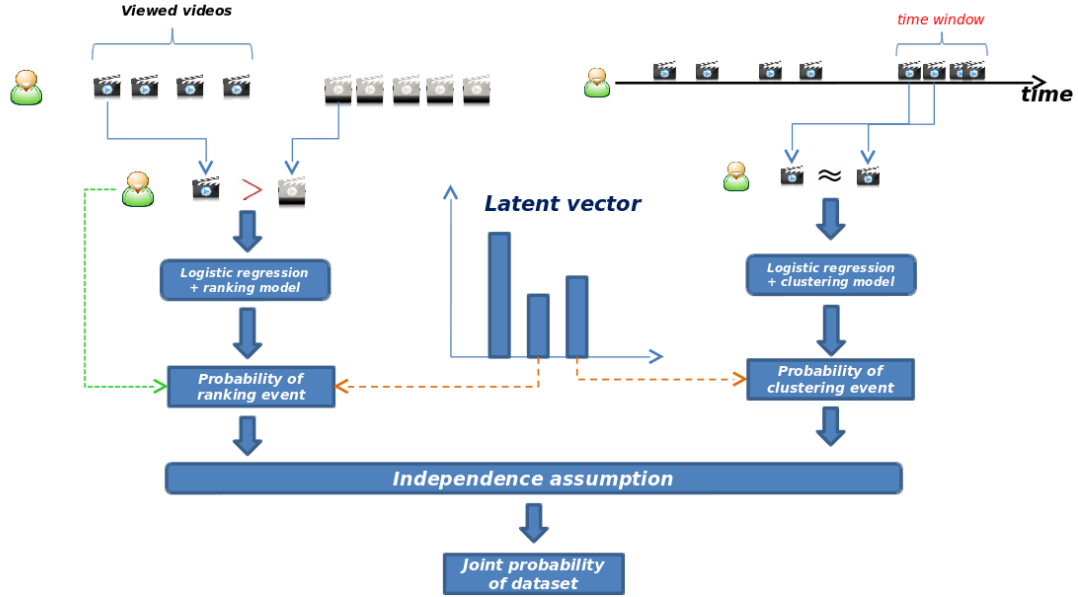


Figure 3.6: integration process

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。

3.5.2 模型描述

图3.7是模型的图形化表示，其中的关键部分是：为了产生两类随机事件，只与一组隐含因子向量随机变量相关。换句话说，一旦隐含用户和视频的隐含因子向量随机变量被确定，两类事件的分布就已经确定。图3.7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。

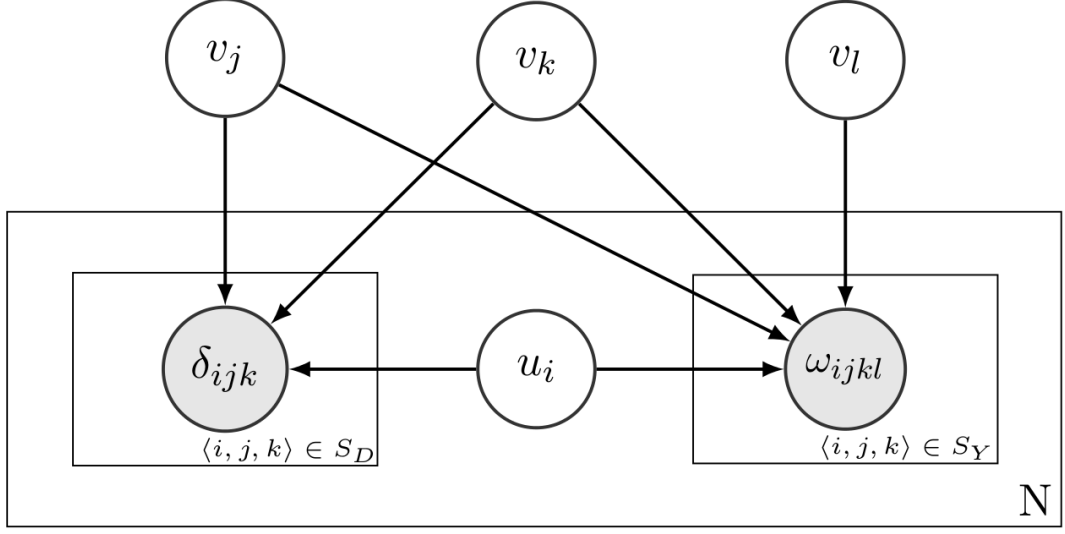


Figure 3.7: graphical model of ClusterRank

框架的生成过程如下：

1. 对每个用户*i*

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

2. 对于每个三元组 $\langle i, j, k \rangle \in S_D$, 采样一个随机变量 $\delta_{ijk} = 1 \sim \text{Bernoulli}(\rho_{ijk})$
其中 ρ_{ijk} 是“用户*i*对*j*的偏好大于对*k*的偏好”的概率：

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

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

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

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

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

通过贝叶斯推导，我们得到如下后验概率：

$$P(\mathbf{U}, \mathbf{V} | \boldsymbol{\delta}, \boldsymbol{\omega}, \lambda_u, \lambda_v) \propto P(\mathbf{U} | \lambda_u) P(\mathbf{V} | \lambda_v) P(\boldsymbol{\delta} | \mathbf{U}, \mathbf{V}) P(\boldsymbol{\omega} | \mathbf{U}, \mathbf{V}) \quad (3.15)$$

使用最大化后验概率方法，我们得到如下目标函数：

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

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

3.5.3 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目标函数，一般做法是一步步分别优化：先优化其中一个目标，然后使用优化后的参数优化另一个目标，如此迭代进行

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

区别对待两类信号可以有如下解释：

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

这样，新的目标函数如下：

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

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

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

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

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

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

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

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

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

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

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

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

实际情况中, 更新会一直持续到目标收敛或者到达最大更新次数.

3.6 实验

3.6.1 实验设计

我们采用的数据集来自Youku, 它是中国最大的视频分享网站. 为了获取足够密集的用户-视频反馈数据, 我们从一段时间内最流行的视频开始抓取, 从这些视频出发, 我们抓取关于这些视频的所有评论, 所有抓取操作都是通过Youku的API完成. 根据80-20[1]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，现在我们将对用户的评论行为作出预测，相比于观看行为，这会是一个更加艰巨的任务。

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

3.6.2 参数设定

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

精准度 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

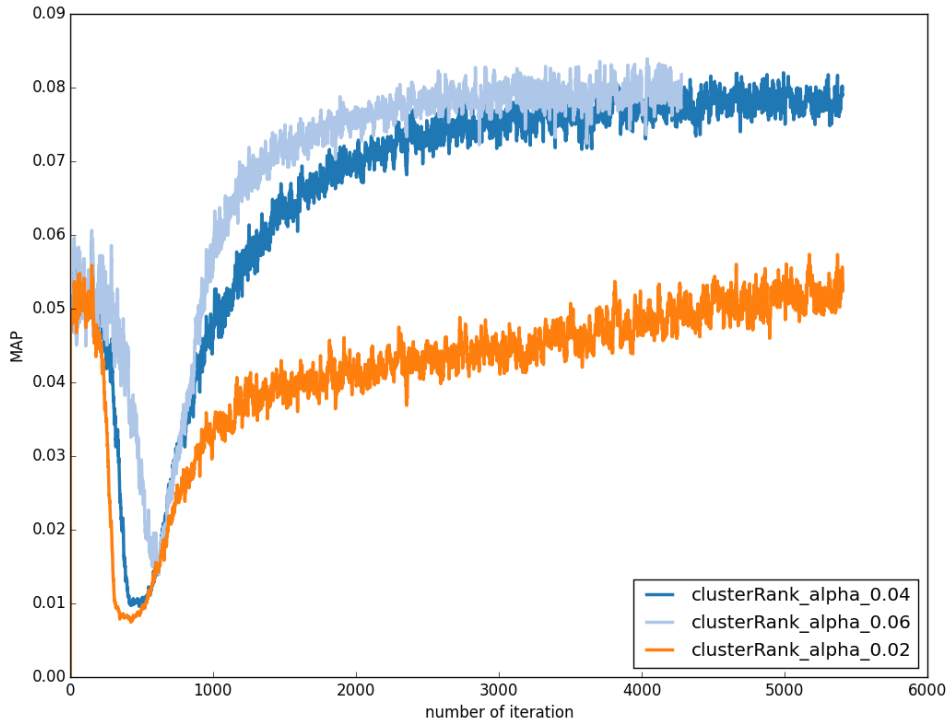
平均准确率 精准度里考虑命中视频顺序的一个版本，信息检索中常用的一个指标

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3.8所示，我们得到了随着 α 的改变得到的平均准确率的变化，横轴代表迭代次数，纵坐标代表平均准确率取值。因为我们使用随机梯度下降优化，结果中存在着上下波动的情况。但是长期来看，结果曲线还是呈现出上升的趋势。结果中的平均准确率先是下降，之后呈现稳定的上升趋势，直到收敛。当我们将 α 从0.04上升到0.06之后结果没有明显的改变，因此我们将设定 $\alpha = 0.06$ 。

3.6.3 结果分析

在图3.9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。

Figure 3.8: MAP for different α

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.

如图3.10, 我们看到对于top-5的推荐结果变得更加有趣. 这一次, 我们的方法在三个指标中的结果的表现都远胜过BPR的结果. 相比top-5与top-10的结果, 我们发现在准确率与召回率上的提升主要的原因是: 原来在5-10名的那些命中视频被准确地推送到了top-5的列表中, 这再次证明了我们的方法在推荐次序上优化的有效性.

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

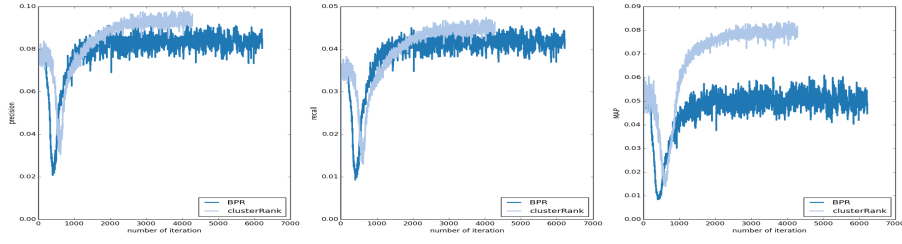


Figure 3.9: Performance for top-10 recommendation

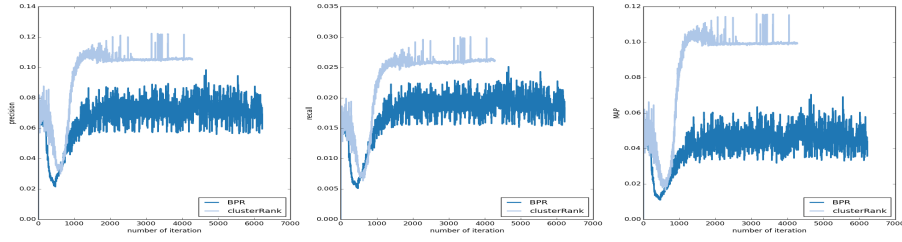


Figure 3.10: Performance for top-5 recommendation

3.7 本章小结

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