计算机科学
COMPUTER SCIENCE

基于差异性汉明距离的变分推荐算法

董家玮, 孙福振, 吴相帅, 吴田慧, 王绍卿

引用本文

董家玮, 孙福振, 吴相帅, 吴田慧, 王绍卿基于差异性汉明距离的变分推荐算法[J]. 计算机科学, 2022, 49(12): 178-184.

DONG Jia-wei, SUN Fu-zhen, WU Xiang-shuai, WU Tian-hui, WANG Shao-qing. Variational Recommendation Algorithm Based on Differential Hamming Distance [J]. Computer Science, 2022, 49(12): 178-184.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220600024

基于差异性汉明距离的变分推荐算法

董家玮　孙福振　吴相帅　吴田慧　王绍卿
山东理工大学计算机科学与技术学院　山东 淄博 255000
(443191260@qq.com)

摘　要　目前基于哈希技术的推荐算法常用汉明距离表示用户和项目哈希码的相似性,但忽略了哈希码中每位的潜在区别信息,为此提出了一个差异性汉明距离,通过考虑哈希码之间的差异性为哈希码赋予位权重;为差异性汉明距离设计了一个变分推荐模型,该模型分为用户哈希组件和项目哈希组件两部分,以变分自编码器结构连接。首先,模型利用编码器为用户和项目生成哈希码,为提高哈希码的鲁棒性,在哈希码中加入高斯噪声。其次,通过差异性汉明距离优化用户和项目哈希码,以最大限度地提高模型重构用户-项目评分的能力。在两个公开的数据集上的实验结果表明,在计算开销不变的前提下与最先进的哈希推荐算法相比,所提模型在 NDCG 上提高了 3.9%,在 MRR 上提高了 4.7%。

关键词:汉明距离;差异性汉明距离;位权重;推荐算法;变分自编码器
中图法分类号　TP391.3

Variational Recommendation Algorithm Based on Differential Hamming Distance

DONG Jia-wei,SUN Fu-zhen,WU Xiang-shuai,WU Tian-hui and WANG Shao-qing
School of Computer Science and Technology,Shandong University of Technology,Zibo,Shandong 255000,China

Abstract　Current recommendation algorithms based on hashing technology commonly uses Hamming distance to indicate the similarity between user hash code and item hash code,while it ignores the potential difference information of each bit dimension. Therefore,this paper proposes a differential Hamming distance,which by calculating the dissimilarity between hash codes to assign bit weights. This paper designs a variational recommendation model for dissimilarity Hamming distance. The model is divided into a user hash component and an item hash component,which are connected by variational autoencoder structure. The model uses encoder to generate hash codes for user and items. In order to improve the robustness of the hash codes,we apply a Gaussian noise to both user and item hash coeds. Besides,the user and item hash codes are optimized by differential Hamming distance to maximize the ability of the model to reconstruct user-item scores. Experiments on benchmark datasets demonstrate that the proposed algorithm VDHR improves 3.9% in NDCG and 4.7% in MRR compared to the state-of-the-art hash recommendation algorithm under the premise of constant computational cost.

Keywords　Hamming distance,Differential Hamming distance,Bit weights,Recommendation algorithm,Variational autoencoder

1　引言

随着在线应用程序的发展,在线应用服务需要处理的信息量激增。推荐系统可以解决信息过载问题,帮助用户更加快捷高效地找到想要的项目,但在用户和项目数量不断增加的情况下,如何准确、有效地匹配项目与潜在用户仍然是当前存在的问题[1-2]。矩阵分解[3]作为协同顾虑技术已经在学术界和工业界取得了巨大成功,通过分解用户-项目评分矩阵,将用户和项目映射到低维的潜在特征空间中,但在大规模数据上推荐时,基于矩阵分解的方法计算成本高且效率低。最近的研究表明,哈希技术可以有效地应对效率问题。它通过将高维空间向量映射成低维的二值编码,使用汉明空间中的汉明距离来衡量向量在原始空间的相似度[4],实现了大数据下的高效存储和快速检索[5-6]。基于哈希的推荐算法将用户和项目数据映射为紧凑的哈希码,通过汉明距离可以计算出用户和项目的相似性,汉明距离越小相似性越高[7]。用户和项目向量的内积被汉明空间的汉明距离所替代,使得在大规模数据下计算复杂度显著地降低[8]。若使用特殊的数据结构先对项目进行索引,计算复杂度可以达到亚线性甚至常数[6,9],同时,低维的哈希码节省了数据存储开销,避免了维度灾难等问题[10]。

目前哈希技术中计算哈希码相似性的方法是汉明距离。

到稿日期:2022-06-02　返修日期:2022-09-02
基金项目:国家自然科学基金(61841602);山东省自然科学基金(ZR2020MF147)
This work was supported by the National Natural Science Foundation of China(61841602) and Natural Science Foundation of Shandong Province,China(ZR2020MF147).
通信作者:孙福振(sunfuzhen@sdut.edu.cn)

根据汉明距离的定义,原始数据相似性对应哈希码位值不同的个数,但是汉明距离中哈希码的位权重都是相等的,哈希码每位的底层属性不同导致其重要性可能不同。为此我们提出了一个差异性汉明距离,通过把项目哈希码映射到用户哈希码上,获得用户哈希码在项目哈希码的位权重。与汉明距离相比,差异性汉明距离不考虑每位表示的正偏好或负偏好,而是考虑每位表示是否重要。为了将差异性汉明距离应用在推荐算法上,我们提出了一个基于差异性汉明距离的变分推荐模型(Variational autoencoder with Differential Hamming distance for Recommendation,VDHR),来学习基于差异性汉明距离优化的哈希码。模型由两个组件构成,分别是用户哈希组件和项目哈希组件。它们以变分自编码器结构进行连接,其中项目哈希组件直接通过项目评论生成项目哈希码,可以最大限度地提高重构原始评论的能力。用户哈希组件通过学习用户嵌入矩阵生成用户哈希码,将用户哈希码与项目哈希码进行联合优化,从而优化观察到的用户-项目评分对数似然,并在公开数据集 Amazon 和 Yelp 上设计实验和分析算法的推荐性能。

2 相关工作

2.1 变分自编码器架构

早期提出基于哈希的协同过滤方法是将哈希码的学习过程分为两个独立的阶段:实值优化和二进制量化[7],但两阶段方法存在较大的量化损失,导致推荐算法性能降低。最近的研究通过两种直接学习哈希码的方式来解决二阶段量化损失问题,一种是通过定义松弛变量来保持哈希码位平衡和去相关约束[11],另一种是使用自编码器以端到端的方式学习哈希码[12-13]。

基于哈希的协同过滤将用户和项目映射为用户哈希码 $z_u\in\{-1,1\}^m$ 和项目哈希码 $z_i\in\{-1,1\}^m$,m 表示哈希码的长度。在哈希算法中,常通过计算用户和特定项目的汉明距离来估计它们的相似性,具体表达式如下:

$$H(z_u,z_i)=\sum_{j=1}^{m}1_{[z_u^{(j)}\neq z_i^{(j)}]}=\mathrm{SUM}(z_u\mathrm{XOR}z_i) \tag{1}$$

从上式可知汉明距离是计算两个哈希码对应位值不同的个数,具体的操作是计算哈希码位异或的总和。而用户和项目的内积可以由汉明距离代替:

$$z_u^{\mathrm{T}}z_i=m-2H(z_u,z_i) \tag{2}$$

2.2 差异性汉明距离

近年来,研究者们提出了很多基于加权汉明距离的哈希排序方法。Zhang 等提出了一种加权汉明距离排序(Weighted hamming distance Ranking,WhRank)算法,哈希码中每位的鉴别能力代表哈希码的位权重,哈希码之间的距离表示为每个哈希码位乘以位权重的总和[14]。Liu 等提出了一种协同过滤的组合编码(Compositional Coding for Collaborative Filtering,CCCF),其利用实值向量的准确性和二值编码的效率来表示用户和项目[15]。但是上述方法存在一个共同的问题,即计算哈希码位权重需要增加额外的存储需求,导致计算效率明显不如汉明距离高。针对上述问题,我们提出了一个差异性汉明距离,在不降低效率的情况下提高计算哈希码位权重的质量。

假设 V 是任意域 F 上的向量空间,设 $\|\cdot\|:V\to R$ 是 V 上的汉明范数,定义 $\vec{u}$ 和 $\vec{i}$ 之间的差异性汉明距离为:

$$DH(\vec{u},\vec{i})=\|\vec{u}-P_u(\vec{i})\| \tag{3}$$

与式(1)类似,u 和 i 相似度越高,它们的差异性汉明距离 $DH(\vec{u},\vec{i})$ 则越小。用 $P_{z_u}(z_i)$ 表示将项目哈希码 z_i 映射到用户哈希码 z_u,推导出差异性汉明距离 $DH(z_u,z_i)$ 的定义:

$$P_{z_u}(z_i)=z_u\ \mathrm{AND}\ z_i \tag{4}$$

$$\begin{aligned}DH(z_u,z_i)&=\|z_u-P_{z_u}(z_i)\|\\&=\mathrm{SUM}(z_u\ \mathrm{XOR}(z_u\ \mathrm{AND}\ z_i))\end{aligned} \tag{5}$$

式(5)中将用户哈希码 z_u 映射到项目哈希码 z_i,表示 z_u 对 z_i 进行哈希码位加权。将用户哈希码 z_u 中为 -1 的位定义为哈希码的不重要位。

与汉明距离相比,差异性汉明距离并不是将哈希码的每个维度定义为正偏好或负偏好,而是定义项目的哪些属性维度对用户来说比较重要,这种方式可以产生特定用户的项目哈希码。

3 基于差异性汉明距离的变分推荐算法

图 1 给出了本文提出的 VDHR 模型的整体结构,具体算法流程为:首先将用户实值矩阵和项目评论输入到模型中,通过嵌入层得到用户和项目的连续嵌入;然后将用户和项目的连续嵌入通过共同的变分自编码器得到对应的用户和项目哈希码,为了提高哈希码的鲁棒性和通用性,在解码前对哈希码加入高斯噪声;最后通过差异性汉明距离优化学习到用户和项目哈希码,并在汉明空间计算相似度以进行推荐。

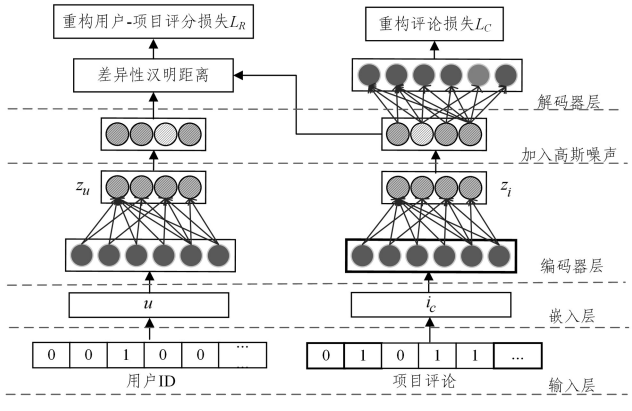


图 1 VDHR 框架

Fig. 1 Framework of VDHR

VDHR 的整体结构包含两个组件,分别为用户哈希组件和项目哈希组件。项目哈希组件直接学习每个项目相关的评论特征得到项目哈希码,项目哈希组件包括两个优化目标:1)通过差异性汉明距离最大限度地提高观察到的用户-项目评分的似然性;2)重建原始评论特征的无监督目标。通过这种设计,VDHR 可以生成更高质量的项目哈希码,更好地表达已观察到的项目评论。用户哈希组件仅通过差异性汉明距离最大化观察到的用户-项目评分的似然性来直接学习用户哈希码。观察到的用户-项目评分的似然性会影响用户和项目哈希码的优化。

3.1 变分自编码器架构

变分自编码器将用户和项目的似然函数定义为：

$$p(u)=\prod_{i\in \boldsymbol{I}_u} p(\boldsymbol{R}_{u,i}) \tag{6}$$

$$p(i)=p(c_i)+\prod_{u\in \boldsymbol{U}_i} p(\boldsymbol{R}_{u,i}) \tag{7}$$

其中，I_u 是用户 u 参与评分的项目集合，U_i 是已经给项目 i 评分的用户合集，$p(c_i)$ 是可以观察到的项目评论似然性，c_i 表示每个项目相关的评论特征。为了最大化用户和项目的似然函数，需最大化观察到的用户-项目评分似然性 $p(\boldsymbol{R}_{u,i})$ 和项目评论似然性 $p(c_i)$。观察到的用户-项目评分和项目评论的对数似然函数为：

$$\log p(\boldsymbol{R}_{u,i})=\log \sum_{z_1,z_i\in\{-1,1\}^m} p(\boldsymbol{R}_{u,i}\mid z_u,z_i)p(z_u)p(z_i) \tag{8}$$

$$\log p(c_i)=\log \sum_{z_i\in\{-1,1\}^m} p(c_i\mid z_i)p(z_i) \tag{9}$$

用户和项目哈希码通过 m 次连续的伯努利试验进行采样得到。$p(z_u)$ 和 $p(z_i)$ 具体的计算式如下：

$$p(z)=\prod_{j=1}^{m} p^{x_j}(1-p)^{1-x_j},x_j=1(z^{(j)}>0) \tag{10}$$

其中，$z^{(j)}$ 表示哈希码的第 j 位。直接优化对数似然函数是很困难的，因此推出一个对数似然函数下限，将用户和项目哈希码的近似后验分布定义为 $q_\theta(z_u\mid u)$ 和 $q_\psi(z_i\mid i)$，θ 和 ψ 是用户和项目的参数集，近似后验分布重写成期望项，最大化詹森不等式来获得观察到的用户-项目评分和项目评论的对数似然函数下限：

$$\log p(\boldsymbol{R}_{u,i})\geqslant E_{q_\psi,q_\theta}[\log[p(\boldsymbol{R}_{u,i}\mid z_u,z_i)]+\log p(z_u)-\log q_\theta(z_u\mid u)+\log p(z_i)-\log q_\psi(z_i\mid i)] \tag{11}$$

$$\log p(c_i)\geqslant E_{q_\psi}[\log p(c_i\mid z_i)+\log p(z_i)-\log q_\psi(z_i\mid c_i)] \tag{12}$$

z_u 和 z_i 通过伯努利试验独立采样得到，$q_\theta(z_u\mid u)$ 和 $q_\psi(z_i\mid i)$ 独立同分布，重写对数似然函数的下限：

$$\log p(\boldsymbol{R}_{u,i})\geqslant E_{q_\psi,q_\theta}[\log p(\boldsymbol{R}_{u,i}\mid z_u,z_i)]-\mathrm{KL}(q_\theta(z_u\mid u)\parallel p(z_u))-\mathrm{KL}(q_\psi(z_i\mid i)\parallel p(z_i)) \tag{13}$$

$$\log p(c_i)\geqslant E_{q_\psi}[\log p(c_i\mid z_i)]-\mathrm{KL}(q_\psi(z_i\mid c_i)\parallel p(z_i)) \tag{14}$$

$\mathrm{KL}(\cdot\parallel\cdot)$ 是 KL 散度(Kullback-Leibler)。KL 散度是对数似然函数的正则项，表示从解码过程中观察到的用户-项目评分或项目评论的程度。

3.2 项目哈希组件

学习到的近似后验分布 q_ψ 和 q_θ 可以被分别看作用户和项目编码器的哈希函数，并通过神经网络公式进行建模。编码器的目标是将用户和项目实值表示转换成对应的 m 位哈希码。

3.2.1 项目编码器

将项目 i 的评论 c_i 进行多层编码，项目哈希码的采样概率如下：

$$l_1=\mathrm{ReLU}(\boldsymbol{W}_1(c_i\odot w_{\mathrm{imp}})+b_1) \tag{15}$$

$$l_2=\mathrm{ReLU}(\boldsymbol{W}_2 l_1+b_2) \tag{16}$$

其中，$\boldsymbol{W}$ 和 b 是模型学习到的权重和偏置，$\odot$ 是元素乘法，w_{imp} 是评论重要性权重。将最后一层 l_2 转换为一个 m 维向量来获得哈希码采样概率：

$$q_\psi(i)=\sigma(\boldsymbol{W}_3 l_2+b_3) \tag{17}$$

其中，σ 是 sigmoid 函数，用于调整编码器输出使其在 0 到 1 之间[16]。伯努利分布中进行采样得到项目哈希码，项目哈希码的第 j 位计算如下：

$$z_i^{(j)}=2\lceil q_\psi(i)^{(j)}-\mu^{(j)}\rceil-1 \tag{18}$$

其中，ψ 是项目编码的参数集。$\mu\in[0,1]^m$ 是有均匀采样的 m 维向量。随机采样 μ 在训练期间可以把项目表示为多个不同的哈希码，但在测试期间需将 μ 设置为 0.5，以确保每个项目都有一个固定的哈希码输出。模型使用一个直通估计器[17]来计算采样哈希码的反向推进的速度。

3.2.2 项目编码器组件

项目解码器的目的是重建原始评论特征，即 $\log p(c_i\mid z_i)$。我们使用 softmax 计算评论对数似然性的总和。

$$\log p(c_i\mid z_i)=\sum_{w\in \boldsymbol{W}_{c_i}} \log \frac{e^{z_i^{\mathrm{T}}(\boldsymbol{E}_{\mathrm{word}}(1_w\odot w_{\mathrm{imp}}))+b_w}}{\sum_{w'\in \boldsymbol{W}} e^{z_i^{\mathrm{T}}(\boldsymbol{E}_{\mathrm{word}}(1_{w'}\odot w_{\mathrm{imp}}))+b_{w'}}} \tag{19}$$

其中，1_w 是项目评论特征 w 的独热编码，W 是项目评论特征向量集合，$\boldsymbol{E}_{\mathrm{word}}\in\boldsymbol{R}^{|\boldsymbol{W}|\times m}$ 是学习到的项目评论嵌入，b_w 是偏置。项目评论重要性权重 w_{imp} 与式(15)相同。

3.3 用户哈希组件

3.3.1 用户编码器组件

用户编码组件和项目编码组件采用的是类似的方式。由于没有用户特征向量，因此用户哈希码通过用户嵌入矩阵学习得到。用户哈希码采样概率如下：

$$q_\theta(u)=\sigma(\boldsymbol{E}_{\mathrm{user}}1_u) \tag{20}$$

其中，$\boldsymbol{E}_{\mathrm{user}}\in\boldsymbol{R}^{|\boldsymbol{U}|\times m}$ 是学习到的用户实值嵌入，1_u 是用户 u 的独热编码。根据 $q_\theta(u)$ 对用户哈希码的每位进行采样，用户哈希码的第 j 位计算如下：

$$z_u^{(j)}=2\lceil q_\theta(u)^{(j)}-\mu^{(j)}\rceil-1 \tag{21}$$

其中，θ 是用户编码的参数集。

3.3.2 用户-项目评分解码

用户-项目评分解码旨在重建原始的用户-项目评分 $\boldsymbol{R}_{u,i}$。首先将用户-项目评分转换为与用户和项目哈希码内积相同的范围：

$$\hat{\boldsymbol{R}}_{u,i}=2m\frac{\boldsymbol{R}_{u,i}}{max\ rating}-m \tag{22}$$

假设用户-项目评分是高斯分布，条件对数似然函数的计算式如下：

$$\log p(\boldsymbol{R}_{u,i}\mid z_i,z_u)=\log N(\hat{\boldsymbol{R}}_{u,i}-DH(z_i^{\mathrm{T}}z_u),\sigma^2) \tag{23}$$

其中，方差 σ^2 是常数，它为所有用户-项目评分提供一个同等的权重。最大化对数似然 $\log p(\boldsymbol{R}_{u,i}\mid z_i,z_u)$ 等价于最小化均方误差(MSE)，即 $\hat{\boldsymbol{R}}_{u,i}-DH(z_i^{\mathrm{T}}z_u)$。

3.4 提高哈希码鲁棒性

研究表明，在哈希码中加入随机噪声可以提高哈希码的鲁棒性和通用性[18-19]，本文提出的模型在解码前对用户和项目哈希码加入高斯噪声，即：

$$noise(z,\sigma^2)=z+\varepsilon\sigma^2,\varepsilon\sim N(0,1) \tag{24}$$

使用方差退火来减小每次训练迭代中 σ^2 的初始值。

3.5 综合损失函数

模型通过最大化式(13)和式(14)中变分下限的组合进行端到端训练：

$$Loss=L_R+\alpha L_c \tag{25}$$

其中，L_R 对应式(13)的下限；L_c 对应式(14)的下限；α 是一个

可调超参数,用于控制解码项目评论的重要性。

4 实验评测

4.1 数据集

本文选用推荐系统上常用的 Yelp 和 Amazon 数据集,训练集、验证集和测试集的比例为 3.5∶1.5∶5。Yelp[1)] 数据集包括用户对酒店、餐厅、景点等地点的评分和评论。Amazon[2)] 数据集是亚马逊 2014 年公开的图书数据集,数据集包含图书信息。Yelp 数据集和 Amazon 数据集中用户的评分范围在 1～5 之间。首先对数据集进行清洗,保留用户对项目的最后一次评分,删除评分数量少于 20 的用户和少于 20 个用户评分的项目,反复清洗直到数据集中所有用户和项目都满足要求。这些数据集的信息描述如表 1 所列。

表 1 预处理后数据集信息

Table 1 Information of pre-processed datasets

数据集	用户数	项目数	评论数	稀疏度/%
Yelp	27147	20266	1293247	99.765
Amazon	35736	38121	1960674	99.856

4.2 评估指标

将模型和基线作为排名任务进行评估,目的是把最相关的项目放在推荐排名的顶部。通过每个用户的测试项目,按照用户和项目哈希码之间的差异性汉明距离排序得到每个用户的固定推荐列表。采用两种常用的评价指标来衡量排名列表的质量。归一化折损累计增益(NDCG)同时包含排名精度和评分的位置;平均倒数排名(MRR)期望评分最高的项目出现在排名顶部,从用户的测试项目列表中计算出具有最高给定评分的最高排名项目。

4.3 基线模型

将 VDHR 模型与使用汉明距离的最先进基线方法进行比较,同时还包括实值方法。实值方法不能直接与哈希方法相比较,因此只作为实值协同过滤的参考点。针对常见的机器字设置哈希码长度为 32 位和 64 位。

NeuHash-CF[20] 神经哈希协同过滤是一种针对冷启动推荐而设计的方法。为避免哈希码的量化误差,使用自编码器来实现端到端的方式直接学习哈希码,提高哈希码的泛化能力。

离散内容感知矩阵分解(Discrete Content-aware Matrix Factorization,DCMF)[21] 是一种内容感知矩阵分解技术。其通过解决多个混合整数的子问题进行优化,并为与每个项目相关的文本中的每个单词学习一个潜在的表示,生成项目的哈希码。

DDL(Discrete Deep Learning)[22] 离散深度学习使用一种交替优化策略来解决多个混合整数子问题。与 DCMF 不同的是,DDL 使用一个深度信念网络来生成项目的哈希码,该网络通过学习将已知项目的评论映射到上一部分生成的哈希码中进行训练。

离散协同过滤(Discrete Collaborative Filtering,DCF)[10] 通过用户项目交互矩阵分解来学习用户和项目哈希码,同时使用位平衡和去相关约束。

为说明哈希码相比实值表示的优越性,本文对比了经典的矩阵分解算法和因式分解机。

因式分解机(Factorization Machines,FM)[23] 在独热编码的用户嵌入表示、独热编码的项目嵌入表示和评论特征的 n 维向量上工作,学习 n 个维度中每个维度的标量权重和偏置。FM 通过计算所有非零项和连接非零项之间的交互权重来估计用户和项目的相关性。

矩阵分解(Matrix Factorization,MF)[3] 学习用户和项目的实值潜在向量,用户和项目向量内积对应用户-项目的相关性。MF 方法和没有任何特征交互的 FM 类似。

这些实值方法在进行推荐时计算成本明显较高。

4.4 实验设计

本文模型采用 TensorFlow 技术完成编写,并在 TESLA P100 上进行训练。使用 Adam 优化器优化模型。模型的学习率设置为 0.0005,batch_size 为 2000,此设置可以使模型在实验中体现出更好的性能。变分自编码器[24] 的编码器层数通常在{1-5}层,每层神经元个数设置为 1000 个,神经网络激活函数使用 Sigmoid。为防止出现过拟合问题,使用 Dropout 保留 80%的神经元。针对哈希码的鲁棒性和通用性,在解码之前对哈希码加入高斯噪声,其中方差最初被设置为 1,每批之后减少 0.01%,式(14)中的 α 设置为 0.001。所有的超参数通过验证集进行调整。

4.5 实验结果分析

4.5.1 模型对比

表 2 和表 3 分别列出了 Yelp 数据集和 Amazon 数据集在 *NDCG*@{2,6,10} 和 *MRR* 的实验对比,其中每列最高的 *NDCG* 和 *MRR* 用粗体表示,实值方法的结果在右上角用标星号的方式与哈希方法进行区分。使用 0.05 水平的双尾 T 检验和 Bonferroni 校正,结果表明,VDHR 明显优于其他基于汉明距离的基线。在 Yelp 数据集上,VDHR 的 *NDCG* 提高了 3.9%,*MRR* 提高了 4.7%;在 Amazon 数据集上,*NDCG* 提高了 3.1%,*MRR* 提高了 3.9%。

本文提出的 VDHR 模型性能明显优于其他基于汉明距离的哈希基线的原因分析如下:DCF 模型首次提出直接学习哈希码,通过平衡和去相关约束提高哈希码的表达能力,但是没有考虑项目的评论信息;DCMF 和 DDL 是离散协同过滤模型,其将评论信息作为辅助信息来提高项目哈希码的表达能力,其中 DDL 使用深度信念网络 DBN 从项目的评论信息中生成项目的特征表示,然而 DBN 的整体结构与优化过程是独立的,这就限制了 DDL 的学习能力;NeuHash-CF 是通过自编码器端到端地生成用户和项目哈希码,自编码器仅仅关注隐层编码的重建能力,但是隐层空间的分布往往是不规律和不均匀的,VDHR 使用变分自编码器构建有规律的隐层空间,可以实现隐层空间随机采样,提高哈希码的鲁棒性和通用性,用差异性汉明距离来弥补汉明距离中的哈希码。

1) https://www.yelp.com/dataset/challenge

2) http://jmcauley.ucsd.edu/data/amazon/

由表2、表3可以发现：1)基于差异性汉明距离优化的哈希模型在每一列都优于基于哈希方法的基线；2)基于差异性汉明距离对MRR的增益要高于NDCG，这意味着差异性汉明距离影响了推荐列表的顶端；3)基于汉明距离的哈希模型基线(DCF，DDL，DCMF，NeuHash-CF)总体上有相似的分数，这表明汉明距离对哈希码的有效性存在上限；4)使用内积的实值向量的FM和MF模型基线优于所有的基于哈希的模型基线，其浮点数的表示能力明显高于哈希码。

表2 Yelp数据集在标准情况下的*NDCG*和*MRR*评估

Table 2 Yelp dataset evaluation of *NDCG* and *MRR* in standard case

	dim=32				*dim*=64			
	NDCG@2	*NDCG*@6	*NDCG*@10	*MRR*	*NDCG*@2	*NDCG*@6	*NDCG*@10	*MRR*
DCF	0.649	0.685	0.738	0.636	0.671	0.700	0.750	0.656
DDL	0.651	0.686	0.739	0.638	0.664	0.698	0.749	0.651
DCMF	0.655	0.691	0.743	0.644	0.671	0.701	0.752	0.660
MF(real-valued)	0.755*	0.763*	0.800*	0.767*	0.755*	0.763*	0.800*	0.767*
FM(real-valued)	0.750*	0.760*	0.798*	0.756*	0.744*	0.755*	0.794*	0.750*
NeuHash-CF	0.659	0.687	0.739	0.645	0.697	0.724	0.770	0.687
VDHR	**0.686**	**0.709**	**0.756**	**0.675**	**0.715**	**0.733**	**0.778**	**0.711**

表3 Amazon数据集在标准情况下的*NDCG*和*MRR*评估

Table 3 Amazon dataset evaluation of *NDCG* and *MRR* in standard case

	dim=32				*dim*=64			
	NDCG@2	*NDCG*@6	*NDCG*@10	*MRR*	*NDCG*@2	*NDCG*@6	*NDCG*@10	*MRR*
DCF	0.759	0.776	0.809	0.751	0.774	0.787	0.818	0.769
DDL	0.748	0.768	0.802	0.741	0.762	0.779	0.811	0.753
DCMF	0.761	0.777	0.810	0.753	0.773	0.788	0.818	0.767
MF(real-valued)	0.824*	0.826*	0.848*	0.826*	0.824*	0.826*	0.848*	0.826*
FM(real-valued)	0.817*	0.819*	0.843*	0.821*	0.813*	0.816*	0.841*	0.815*
NeuHash-CF	0.751	0.767	0.801	0.741	0.778	0.790	0.819	0.770
VDHR	**0.774**	**0.786**	**0.819**	**0.770**	**0.796**	**0.804**	**0.831**	**0.793**

此外还可以发现，随着编码位数的增多，哈希方法性能呈逐渐上升趋势，而实值方法的性能维持不变甚至有下降趋势，其原因主要是实值特征表示较哈希码可以存储更多的信息，更容易出现过拟合现象，因此降低了推荐性能。本文提出的VDHR可以显著缩小哈希方法与实值方法之间的性能差距。

4.5.2 冷启动问题

冷启动问题的数据集设置旨在训练集中未观察到但在测试集中观察到的用户-项目交互。因此对于冷启动的设置，首先根据项目的评分数量进行排序，按照5:5的比例分成训练集和测试集，保证每个集合中有大约相同数量的项目和相似数量的评分，然后使用15%的训练集作为验证集来调整超参数。训练集通过删掉用户-项目交互来构建冷启动环境，具体的Yelp数据集和Amazon数据集的结果如表4、表5所列。

由表4和表5可以发现，本文提出的VDHR在冷启动问题上的结果明显优于其他基于哈希的基线。在Yelp数据集上*NDCG*提高了5.9%，*MRR*提高了7.5%；在Amazon数据集上模型的提升不如在Yelp数据集上明显，提升的范围在0.3%～1.6%之间。

从实验中还能看出32位哈希码的性能提升甚至优于64位哈希码的性能提升，证明了VDHR通过差异性汉明距离优化生成哈希码包含信息的能力。其中在Amazon数据集下64位哈希码的表现甚至优于实值基线FM。在Yelp数据集上，VDHR和FM在NDCG上的差距在0.01内，说明哈希码的表达能力在冷启动问题上与实值方法非常接近。

表4 Yelp数据集在冷启动情况下*NDCG*和*MRR*评估

Table 4 Yelp dataset evaluation of *NDCG* and *MRR* in the cold start case

	dim=32				*dim*=64			
	NDCG@2	*NDCG*@6	*NDCG*@10	*MRR*	*NDCG*@2	*NDCG*@6	*NDCG*@10	*MRR*
DDL	0.579	0.622	0.681	0.562	0.612	0.646	0.700	0.604
DCMF	0.617	0.655	0.709	0.604	0.626	0.664	0.717	0.612
FM(real-valued)	0.724*	0.744*	0.785*	0.722*	0.719*	0.740*	0.781*	0.717*
NeuHash-CF	0.640	0.676	0.728	0.624	0.680	0.714	0.761	0.670
VDHR	**0.678**	**0.710**	**0.757**	**0.671**	**0.706**	**0.731**	**0.774**	**0.704**

表5 Amazon数据集在冷启动情况下*NDCG*和*MRR*评估

Table 5 Amazon dataset evaluation of NDCG and *MRR* in the cold start case

	dim=32				*dim*=64			
	NDCG@2	*NDCG*@6	*NDCG*@10	*MRR*	*NDCG*@2	*NDCG*@6	*NDCG*@10	*MRR*
DDL	0.705	0.729	0.767	0.694	0.705	0.727	0.766	0.694
DCMF	0.729	0.749	0.784	0.721	0.733	0.752	0.786	0.726
FM(real-valued)	0.785*	0.793*	0.821*	0.784*	0.780*	0.790*	0.819*	0.780*
NeuHash-CF	0.759	0.776	0.806	0.751	0.787	0.799	0.826	0.784
VDHR	**0.766**	**0.780**	**0.810**	**0.763**	**0.792**	**0.802**	**0.828**	**0.791**

4.5.3 差异性汉明距离对模型的影响

为了验证本文提出的差异性汉明距离对模型性能的影响，与基于汉明距离的经典算法进行对比，我们设置DCF变体DCF_{DH}为使用差异性汉明距离的离散协同过滤，VDHR变

体 $VDHR_{no.DH}$ 为使用汉明距离的变分推荐模型。

从表 6 可以发现使用差异性汉明距离比汉明距离在 *NDCG*@10 和 *MRR* 上有明显提升,其中 *MRR* 的增长要高于 *NDCG*@10,这表明差异性汉明距离影响推荐列表的顶端,但 *NDCG*@10 和 *MRR* 提升的程度不大表明哈希码表示的能力相较于实值表示是有限的。

表 6 Yelp 数据集上 *NDCG*@10 和 *MRR* 评估

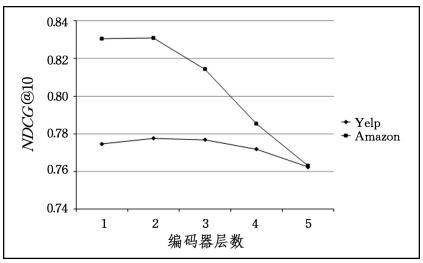
Table 6 Yelp dataset evaluation of *NDCG*@10 and *MRR*

dim=32	*NDCG*@10	*MRR*
DCF	0.738	0.636
DCF_{DH}	0.746	0.654
$VDHR_{no.DH}$	0.741	0.647
VDHR	**0.756**	**0.675**

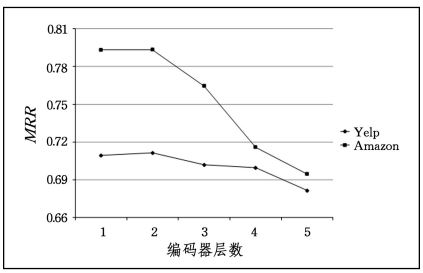
4.5.4 编码器层数对哈希码的影响

编码器本质上是由浅层的神经网络模型提取抽象的特征表示,进而约束哈希码的学习,但神经网络要具有一定的"容量",随着神经网络层数的增加,可以利用有限的神经单元训练学习到数据更深层次的特征,这些抽象的特征能够对模型相关参数进行更本质的描述。

由图 2 可以发现编码器中神经网络层数的增多并不能为模型带来更好的效果,在 Amazon 数据集上表现最为明显,当达到 3 层以上时,模型性能大幅度下降;当编码器层数达到 5 层时,模型出现过拟合。原因可能是哈希码加入高斯噪声后导致模型过拟合。网络层数通过实验来确定,考虑到网络训练成本,在保证模型预测效果的同时,选取层数最少的网络层。根据实验发现,编码器有 2 层神经网络时模型效果最好。



(a) Yelp 数据集和 Amazon 数据集在不同编码器层数上的 *NDCG*@10 评估



(b) Yelp 数据集和 Amazon 数据集在不同编码器层数上的 *MRR* 表现

图 2 Yelp 数据集和 Amazon 数据集在不同编码器层数上的 *NDCG*@10 和 *MRR* 评估

Fig. 2 Yelp dataset and Amazon dataset evaluation of *NDCG*@10 and *MRR* with different number of encoder layers

4.5.5 哈希方法和非哈希方法效率对比

为体现哈希码与实值向量相比的优势,实验设计当存在 100000 个用户和{10000,100000,1000000}个项目时,汉明空间和实值方法的计算效率。实验为用户和项目随机生成哈希码和实值向量,并计算在汉明空间下每个用户到所有项目的距离以及实值用户和项目的实值内积所花费时间。实验使用的硬件设施是 64 位指令集计算机,产生长度为 64 的哈希码和实值向量。

图 3 给出了重复 10 次的平均运行时间,由于实值内积方法在项目数为 1000000 时的时间为 4154.512 s,影响了图 3 中其他数据的展示,因此遮盖掉了部分图样,以更好地展示其他数据。可以发现,汉明距离和本文提出的差异性汉明距离在时间花销上明显优于实值内积方法,它们之间的差距可以忽略不计。实值方法存在大量的内积计算,计算时间花销和特征数成 4 次方,在大规模数据的情况下效率过低。因此,差异性汉明距离增加的任务计算时间花销与实值方法相比可以忽略不计,并且效率有所提高。

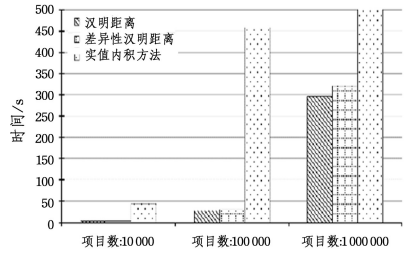


图 3 3 种方法运行时间开销

Fig. 3 Runtime overhead of three methods

结束语 本文发现汉明距离在计算相似性上对哈希码每个位权重保持一致,没有考虑哈希码位上的底层属性,因此提出了差异性汉明距离计算哈希码位权重来区分这种底层属性。本文设计了一个变分推荐模型,针对所提出的差异性汉明距离进行优化。模型由两个联合组件构成,用户和项目哈希码从各组件学习到的伯努利分布中采样得到,增强了端到端的可训练性。在协同过滤实验中证明了差异性汉明距离优化的模型,可以在不降低效率和没有额外存储开销的情况下有效地提高精度,在 Yelp 和 Amazon 数据集中 *NDCG* 提高了 3.9%,*MRR* 提高了 4.7%。

下一步工作是在目前的模型上接受更丰富的用户和项目表示,进一步提高模型推荐效果,主要包括两个方面:1)对评论进行更细致的处理,如考虑时序、情感分析等;2)本文只考虑了用户的 ID 属性,未来将融合用户的辅助信息,如社会关系属性、地理属性等。

参 考 文 献

[1] WANG Z S,LI Q,WANG J,et al. Real-Time Personalized Recommendation Based on Implicit User Feedback Data Stream [J]. Chinese Journal of Computers,2016,39(1):53-64.

[2] HUANG C L,LU Y X. Research on Hybrid Music Recommendation Algorithm based on Collaborative Filtering and Tags[J]. Software Engineering,2021,24(4):10-14.

[3] YEHUDA K,ROBERT B,CHRIS V. Matrix factorization techniques for recommender systems[J]. Computer,2009,42(8):30-37.

[4] LI H Q,WANG Y X,CHEN Z D,et al. Ranking-Based Supervised Discrete Cross-Modal Hashing[J]. Chinese Journal of

Computers, 2021, 44(8): 1620-1635.
[5] WU Z B, YU J Q, HE Y F, et al. Multi-level Semantic Binary Descriptor for Image Retrieval[J]. Chinese Journal of Computers, 2020, 43(9): 1641-1655.
[6] ZHANG Z W, WANG Q F, RUAN L Y, et al. Preference preserving hashing for efficient recommendation[C]// Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. 2014: 183-192.
[7] ZHOU K, ZHA H Y. Learning binary codes for collaborative filtering[C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2012: 498-506.
[8] SALAKHUTDINOV R, HINTON G. Semantic hashing [J]. International Journal of Approximate Reasoning, 2009, 50(7): 969-978.
[9] ZOU A, HAO W N, JIN D W, et al. Study on Text Retrieval Based on Pre-training and Deep Hash[J]. Computer Science, 2021, 48(11): 300-306.
[10] CHEN Q, DAI Y W, LIU G J. Research on KPI anomaly detection model for intelligent operation and maintenance[J]. Journal of Chongqing University of Technology(Natural Science), 2022, 36(6): 181-188.
[11] ZHANG H W, SHEN F M, LIU W, et al. Discrete Collaborative Filtering[C]// Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA, 2016: 325-334.
[12] HUANG X L, CHEN C M, LIU G H. A hybrid second-order total variational noise reduction method for radiation-resistant images[J]. Journal of Chongqing University of Technology (Natural Science), 2022, 34(4): 585-594.
[13] CHAIDAROON S, FANG Y. Variational deep semantic hashing for text documents[C]// Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017: 75-84.
[14] ZHANG L, ZHANG Y D, TANG J H, et al. Binary code ranking with weighted Hamming distance[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2013: 1586-1593.
[15] LIU C H, LU T, WANG X, et al. Compositional Coding for Collaborative Filtering[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 145-154.
[16] CHEN W J. An Adaptive Approach for Knowledge Representation Fused with Topic Feature[J]. Computer Engineering, 2021, 47(1): 87-93, 100.
[17] OOSTERHUIS H, RIJKE M. Unifying Online and Counterfactual Learning to Rank: A Novel Counterfactual Estimator that Effectively Utilizes Online Interventions[C]// Proceedings of the 14th ACM International Conference on Web Search and Data Mining(WSDM '21). Association for Computing Machinery, New York, USA, 2021: 463-471.
[18] CHAIDAROON S, EBESU T, FANG Y. Deep Semantic Text Hashing with Weak Supervision[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. 2018: 1109-1112.
[19] ZHAO B Y, WANG L S, ZHANG M l, et al. Random Plaintext Collision Attack Against AES Algorithm with Reused Masks [J]. Computer Engineering, 2022, 48(6): 139-145, 153.
[20] HANSEN C, HANSEN C, SIMONSEN J G, et al. Content-aware Neural Hashing for Cold-start Recommendation[C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, USA, 2020: 971-980.
[21] LIAN D F, LIU R, GE Y, et al. Discrete Content-aware Matrix Factorization[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 325-334.
[22] ZHANG Y, YIN H Z, HUANG Z, etal. Discrete Deep Learning for Fast Content-Aware Recommendation[C]// ACM International Conference on Web Search and Data Mining. 2018: 717-726.
[23] SUN Y, PAN J W, ZHANG A, et al. FM2: Field-matrixed Factorization Machines for Recommender Systems[C]// Proceedings of the Web Conference 2021(WWW '21). Association for Computing Machinery, New York, USA, 2021: 2828-2837.
[24] GUO F Q, MENG F R, WANG Z X. Rumor Stance Classification Algorithm Based on Variational Auto-Encoder[J]. Computer Engineering, 2022, 48(2): 99-105.

DONG Jia-wei, born in 1998, postgraduate, is a member of China Computer Federation. His main research interests include recommender systems and so on.

SUN Fu-zhen, born in 1978, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include computer vision, data mining and data analysis, etc.

(责任编辑:何杨)