



# 计算机科学

COMPUTER SCIENCE

## 基于变分持续贝叶斯元学习的推荐算法

朱文韬, 刘威, 梁上松, 朱怀杰, 印鉴

### 引用本文

朱文韬, 刘威, 梁上松, 朱怀杰, 印鉴 [基于变分持续贝叶斯元学习的推荐算法](#) [J]. 计算机科学, 2023, 50(7): 66-71.

ZHU Wentao, LIU Wei, LIANG Shangsong, ZHU Huaijie, YIN Jian. [Variational Continuous Bayesian Meta-learning Based Algorithm for Recommendation](#) [J]. Computer Science, 2023, 50(7): 66-71.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

### Similar articles recommended (Please use Firefox or IE to view the article)

#### [基于约束图正则的块稀疏对称非负矩阵分解](#)

Block Sparse Symmetric Nonnegative Matrix Factorization Based on Constrained Graph Regularization

计算机科学, 2023, 50(7): 89-97. <https://doi.org/10.11896/jsjcx.220500050>

#### [基于多目特征交叉的服务推荐算法](#)

Service Recommendation Algorithm Based on Multi-features Crossing

计算机科学, 2023, 50(6A): 210800242-7. <https://doi.org/10.11896/jsjcx.210800242>

#### [受人脑中记忆机制启发的增量目标检测方法](#)

Incremental Object Detection Inspired by Memory Mechanisms in Brain

计算机科学, 2023, 50(2): 267-274. <https://doi.org/10.11896/jsjcx.220900212>

#### [基于差异性汉明距离的变分推荐算法](#)

Variational Recommendation Algorithm Based on Differential Hamming Distance

计算机科学, 2022, 49(12): 178-184. <https://doi.org/10.11896/jsjcx.220600024>

#### [基于Apriori算法填充数据及改进相似度的推荐算法](#)

Recommendation Algorithm Based on Apriori Algorithm and Improved Similarity

计算机科学, 2022, 49(11A): 211000005-5. <https://doi.org/10.11896/jsjcx.211000005>

# 基于变分持续贝叶斯元学习的推荐算法

朱文韬<sup>1,3</sup> 刘威<sup>1,3</sup> 梁上松<sup>1,3</sup> 朱怀杰<sup>2,3</sup> 印鉴<sup>2,3</sup>

1 中山大学计算机学院 广州 511436

2 中山大学人工智能学院 广东 珠海 519082

3 广东省大数据分析处理重点实验室 广州 510006

(zhuwt8@mail2.sysu.edu.cn)

**摘要** 元学习方法近年被引入推荐系统以缓解冷启动问题。现有元学习算法只能提高算法处理一组静态分布的数据集(任务)的能力。当面对多个服从非平稳分布的数据集时,现有模型往往会出现负知识转移以及灾难性遗忘问题,导致算法推荐性能大幅下降。探索了基于变分持续贝叶斯元学习(Variational Continuous Bayesian Meta-Learning, VC-BML)的推荐算法。首先,算法假设元参数服从动态混合高斯模型,使其具有更大的参数空间,提高了模型适应不同任务的能力,缓解了负知识转移问题。然后,VC-BML的任务集群数量由中国餐馆过程(Chinese Restaurant Process, CRP)来灵活确定,使得模型在不同的混合分量中存储不同任务分布的知识,并在类似任务出现时调用这些知识,有助于缓解传统算法中的灾难性遗忘问题。为了估计模型参数的后验概率,算法采用了一种更稳健的结构化变分推理方法来近似后验值,以避免遗忘知识。最后,VC-BML在4个非平稳分布的数据集上的表现均优于基准算法。与基于点估计的基准算法相比,VC-BML提高了模型的稳健性,有助于缓解灾难性遗忘问题。

**关键词:** 推荐算法;冷启动;元学习;动态混合高斯模型

**中图法分类号** TP391

## Variational Continuous Bayesian Meta-learning Based Algorithm for Recommendation

ZHU Wentao<sup>1,3</sup>, LIU Wei<sup>1,3</sup>, LIANG Shangsong<sup>1,3</sup>, ZHU Huaijie<sup>2,3</sup> and YIN Jian<sup>2,3</sup>

1 School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 511436, China

2 School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, Guangdong 519082, China

3 Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China

**Abstract** Meta-learning methods have been introduced into recommendation algorithms in recent years to alleviate the problem of cold start. The existing meta-learning algorithms can only improve the ability of the algorithm to deal with a set of statically distributed data sets(tasks). When faced with multiple data sets subject to non-stationary distribution, the existing models often have negative knowledge transfer and catastrophic forgetting problems, resulting in a significant decline in algorithm recommendation performance. This paper explores a recommendation algorithm based on variational continuous Bayesian Meta-learning (VC-BML). Firstly, the algorithm assumes that the meta parameters follow the dynamic mixed Gaussian model, which makes it have a larger parameter space, improves the ability of the model to adapt to different tasks, and alleviates the problem of negative knowledge transfer. Then, the number of task clusters in VC-BML is flexibly determined by Chinese restaurant process(CRP), which enables the model to store knowledge of different task distributions in different mixed components and invoke this knowledge when similar tasks occur, helping to alleviate the catastrophic forgetting problem in traditional algorithms. To estimate the posterior probabilities of model parameters, a more robust structured variational reasoning method is used to approximate the posterior values to avoid forgetting knowledge. Finally, VC-BML outperforms the baselines on all four datasets with non-stationary distributions. Compared with point-based estimation methods, VC-BML improves the robustness of the model and helps alleviate the catastrophic forgetting problem.

**Keywords** Recommendation algorithm, Cold-start problem, Meta-learning, Dynamic Gaussian mixture model

到稿日期:2022-09-24 返修日期:2023-02-07

基金项目:国家自然科学基金(U1911203, 61902439, 61902438, 62002396);广东省基础与应用基础研究基金(2021A1515011902, 2020A1515011251, 2019A1515011159, 2019A1515011704)

This work was supported by the National Natural Science Foundation of China(U1911203, 61902439, 61902438, 62002396) and Guangdong Basic and Applied Basic Research Foundation(2021A1515011902, 2020A1515011251, 2019A1515011159, 2019A1515011704).

通信作者:刘威(liuw259@mail.sysu.edu.cn)

## 1 引言

推荐算法能够根据海量的用户数据,准确分析用户的喜好,迅速找到并向用户推荐其感兴趣的信息,在电子商务、短视频推送等各种领域都得到了非常广泛的运用。目前,推荐算法主要使用基于内容推荐、协同过滤这两种方法。基于内容的推荐算法通过用户的偏好记录和条目内容,预测用户对新项目的喜好。而协同过滤基于特定用户的行为,寻找相似用户的行为喜好模式,并向该用户进行推荐。但是由于用户交互数据过于稀疏和隐私保护等问题,很难收集足够的用户个人信息,这些算法可能会出现用户冷启动等问题,即算法无法向仅与少量条目进行交互的新用户进行准确的推荐。如果新用户最初使用系统时无法获得令其满意的推荐,他们很容易放弃该系统。为避免此类冷启动问题,许多系统<sup>[1]</sup>会在最初向新用户展示目前流行的条目,使用户从中挑选出他们感兴趣的部分,再根据他们的选择进行推荐。但是如果用户只与少数初始条目进行交互,冷启动问题依然存在<sup>[2-4]</sup>。

为此 Lee 等提出了 MeLU 方法<sup>[5]</sup>,尝试构建了一个基于元学习的推荐算法,其可以仅凭少量数据来预测用户的偏好,提升对新用户的初始推荐性能。但是该算法要求学习的任务都遵循平稳分布,而这种假设在现实中可能是不成立的,后续的任务可能与之前的任务不同,甚至互相矛盾。例如,用于推荐电影的算法无法很好地为用户推荐书籍<sup>[6]</sup>。这类方法存在两个问题:1)模型中的参数可能在处理不同的任务中表现出性能下降的情况,即负知识转移问题;2)为了快速学习不同的任务,模型的参数需要发生大幅度的变化,因此会遗忘之前学习到的知识,即灾难性遗忘问题。因此我们希望能够拓展传统的元学习方法,得到一个性能更优的模型,使其能够处理非平稳分布的任务,避免知识遗忘,适应更多的推荐场景。

根据上述的问题,本文提出了一种基于变分持续贝叶斯学习(VC-BML)的推荐算法,尝试解决 MeLU 中出现的负知识转移以及灾难性遗忘问题。我们的研究有 3 个主要贡献:1)提出了元参数服从动态混合高斯模型(Dynamic Gaussian Mixture Model, DGMM)的假设,相比单一静态的高斯分布,其提高了模型的适应性,其中每个分量与一个任务集群关联,使我们的模型能够适用于更多种类条目的推荐,提高了模型抵抗负知识转移问题的能力;2)使用了结构化变分推理的方法来近似参数的后验分布,与之前的点估计方法相比,其可以量化模型的不确定性,有助于缓解灾难性遗忘问题,提高了模型的稳健性;3)经过实验测试,基于 VC-BML 的推荐算法在 4 个非平稳分布的数据集上的表现均优于对照算法。

## 2 相关工作及理论基础

### 2.1 元学习

元学习(Meta-learning)旨在从训练任务中提取出任务间的共同模式,训练一个能够快速根据少量数据适应新任务的模型<sup>[7]</sup>。其中,基于度量的元学习以及基于记忆的元学习主要用于解决分类问题。相比之下,基于优化的元学习<sup>[8-9]</sup>能够根据每位用户的历史数据向用户提供个性化的推荐算法。

### 2.2 变分持续学习

持续学习旨在解决灾难性遗忘问题,而变分持续学习

(Variational Continuous Learning, VCL)是持续学习中的一种贝叶斯方法,它能够有效地通过后验概率回顾过去的知识,避免历史知识的遗忘。在变分连续学习中,后验分布往往难以处理,需要采用近似的方法。假设对于给定的模型参数  $\theta$  个数数据集都是独立的。参数在第  $t-1$  个数集上的后验概率在第  $t$  个数集  $D_t$  上被作为先验使用。因此,可以使用递归的方法来近似求出参数的后验概率。

$$\begin{aligned} p(\theta_t | D_{1:t}) &\propto p(\theta_t) \prod_{t'=1}^t p(D_{t'} | \theta_t) \\ &\propto p(D_t | \theta_t) p(\theta_t | D_{1:t-1}) \end{aligned}$$

### 2.3 变分持续贝叶斯学习

为了提高元学习模型对多种任务的适应能力,有研究者<sup>[10]</sup>使用动态更新元参数分布的混合,其中每个元参数分布都与一组相似的任务关联,如图 1 所示。假设混合元参数分布为高斯混合模型(Gaussian Mixture Model, GMM),且任务要求 GMM 的参数(均值和方差)能够随着新任务的出现而动态更新。因此,不同于以往实验中的单一静态高斯分布,参数后验概率  $p(\theta_t | D_{1:t})$  服从动态高斯混合模型。相较于单一分布,混合元参数分布提高了元学习处理更多不同任务的能力。设  $z$  为潜在分类变量,表明某个任务所属的任务集群,即在混合分布中选择特定的分量。根据  $z$  的定义,可以重新定义条件概率:

$$\begin{aligned} p(\theta_t | D_t) &= \int p(D_t | \theta_t, z_t) p(z_t) dz_t \\ &= \int \left[ \int p(D_t | \phi_t) p(\phi_t | \theta_t) d\phi_t \right] p(z_t) dz_t \end{aligned}$$

即从  $\theta_t$  的混合分布中选出一个分量分布  $\theta_t^z$ 。在此基础上,得到每个任务的特定参数  $\phi_t$  的分布。

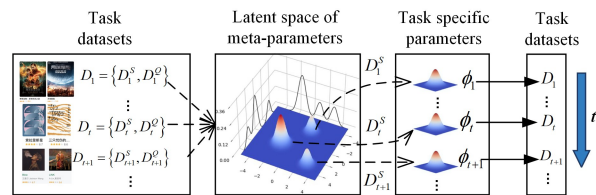


图 1 VC-BML 的框架图

Fig. 1 Block diagram of VC-BML

## 3 基于 VC-BML 的用户偏好估计

### 3.1 用户偏好估计模型

我们提出的用户偏好估计模型如图 2 所示,将用户信息(如年龄、性别等)和条目信息(如发行年份、类别等)作为输入。该模型首先对输入执行嵌入操作,并将嵌入向量串联后使用多层神经网络对决策过程建模,这在最近的有关推荐算法的研究中也得到了广泛的应用<sup>[11-12]</sup>。

由于之前的研究表明嵌入过程能提升推荐算法的性能<sup>[13-14]</sup>,我们利用嵌入过程来提取对估计用户偏好有用的特征。我们的模型使用了文献<sup>[11]</sup>中的方式嵌入用户和条目信息。每一个类目的信息会产生嵌入信息,并使用图 2 中的方式串联。当用户信息类目的数量为  $P$  时,用户  $i$  的嵌入向量  $U_i$  为:  $U_i = [e_{i1}c_{i1}; \dots; e_{iP}c_{iP}]^T$ , 其中  $c_{ip}$  是用户  $i$  的类目信息  $p \in \{1, \dots, P\}$  的  $J$  维独热向量,  $e_{ip}$  表示用户相关类目信息的  $d_c \times d_p$  嵌入矩阵。  $d_c$  和  $d_p$  分别是嵌入维度和用户信息  $p$  的

类目数, 条目信息以相同方式嵌入。  $Q, d_e$  和  $d_q$  分别是条目信息数、嵌入维度和条目信息  $q$  的类目数。

用户偏好估计由模型的决策层和输出层共同完成。用户和条目的嵌入维度可能不同, 因此我们使用了一个  $N$  层全连接神经网络来构造决策层。而输出层可利用决策层给出的信息评估用户的偏好, 可以对条目的评级以及消费可能性等多种方式进行刻画。这些层可表示为:

$$\begin{aligned} \mathbf{x}_0 &= [\mathbf{U}_i; \mathbf{I}_j] \\ &\vdots \\ \mathbf{x}_N &= a(\mathbf{W}_N^T \mathbf{x}_{N-1} + \mathbf{b}_N) \\ \hat{\mathbf{y}}_{ij} &= \sigma(\mathbf{W}_o^T \mathbf{x}_N + \mathbf{b}_o) \end{aligned} \quad (1)$$

其中,  $\mathbf{W}_n$  和  $\mathbf{b}_n$  分别是第  $n$  个决策层的权重矩阵和偏置向量;  $\hat{\mathbf{y}}_{ij}$  是用户  $i$  对条目  $j$  偏好程度的估计;  $a$  和  $\sigma$  分别为决策层和输出层的激活函数, 使用 ReLU<sup>[15]</sup> 作为激活函数  $a$ , 而激活函数  $\sigma$  需要通过估计目标来确定。

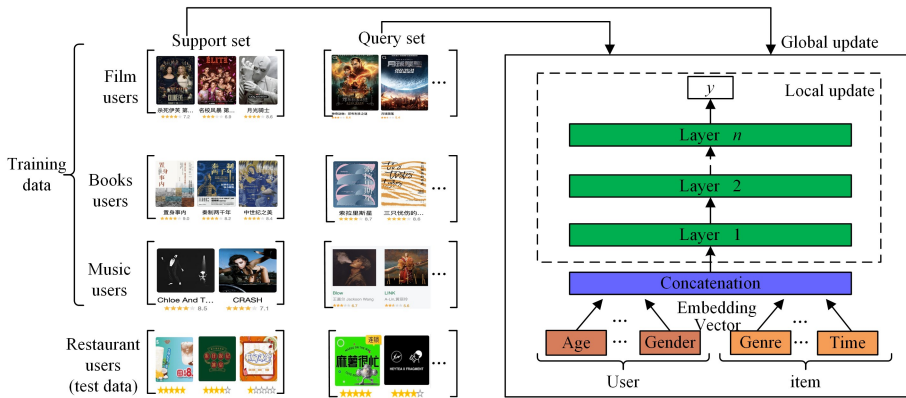


图2 基于 VC-BML 的推荐算法的更新过程

Fig. 2 Update process of VC-BML based recommendation algorithm

此外, 假设模型的元参数服从动态高斯混合模型, 需要利用中国餐馆过程来确定分量分布的数量。由于元参数的参数空间增大了, 我们的模型适应不同任务的能力和抵抗负知识转移问题的能力便大大增强。由于元变量与潜在变量的维度很大, 计算所有变量的精确后验概率是很困难的。为了高效地得到变量的后验概率, 同时量化模型的不确定性, 使用了结构化变分后验来近似准确的后验。此方法比 MeLU 中所使用的点估计方法更加稳健, 在不同任务间传递知识以及缓解灾难性遗忘等方面都具有明显的优越性。准确的后验概率可以通过以下结构化变分后验来近似:

$$q(\phi_i, \theta_i, z_i, \beta | D_i) = q(\phi_i | \theta_i^z, D_i; |\lambda_i) q(\theta_i^z | D_i; \psi_i) q(z_i; \eta_i) \times \prod_{k=1}^{K-1} q(\beta_k; \gamma_k)$$

其中,  $\gamma_k, \eta_i, \psi_i$  和  $\lambda_i$  是 4 个变分分布的参数, 即  $\beta_k, z_i, \theta_i$  和  $\phi_i$  的参数, 共同构成中国餐馆-动态高斯混合模型 CRP-DGMM。

算法 1 展示了完整的生成个性化用户偏好估计模型的过程。第一步, 模型中的参数被随机初始化, 假设元参数服从动态高斯混合模型。由于在局部更新的过程中仅有式(1)中的参数被更新, 因此式(1)中的参数需要被单独表示。第二步, 模型会随机抽取一批用户。第三步, 模型通过反向传播的方式局部地更新这批用户的参数, 损失函数为:

$$\mathcal{L}_i = \frac{1}{|\mathbf{H}_i|} \sum_{j \in \mathbf{H}_i} (\mathbf{y}_{ij} - \hat{\mathbf{y}}_{ij})^2$$

### 3.2 基于 VC-BML 的用户偏好估计

针对已有推荐算法中存在的一些不足<sup>[10-11]</sup>, 我们结合 VC-BML 实现用户偏好估计, 使得在仅有少量的用户项交互数据时, 也能较为准确地估计出用户偏好, 且尽量使我们的模型能够适用于更多不同的任务, 尝试避免负知识转移以及灾难性遗忘。图 2 给出了所提模型如何利用系统中的历史交互数据学习公共知识。模型使用了用户的历史交互数据和条目偏好作为局部更新的支持集, 即在局部更新中, 为了更准确地反映特定用户的偏好, 仅仅利用特定用户的历史数据来更新决策层和输出层的参数(权重矩阵和偏置向量)。在这个过程中, 为了使得模型能够适用于多种任务, 不会更新用户和条目的嵌入向量。直观而言, 模型假设在局部更新时, 用户和条目都不会发生变化, 仅仅用户的想法会在与条目进行交互时发生变化。在对所有用户的局部更新都结束后, 我们再根据查询集对所有层进行全局更新。

其中,  $\mathbf{H}_i = \{j | \text{用户 } i \text{ 交互过的条目 } j\}$  是用户  $i$  浏览过的条目的合集, 即支持集;  $\mathbf{y}_{ij}$  是用户  $i$  实际对  $j$  的偏好程度。这里的局部更新可被认为是个性化迭代的过程, 可以多次重复。最后, 模型根据相应查询集的损失函数  $\mathcal{L}_i'$  全局更新  $\phi_i^k$  和  $\phi_i'^k$ 。这个过程旨在通过对所有用户进行局部更新, 找到能使模型达到较好推荐结果的理想参数。算法会重复这个过程, 直到  $\phi_i^k$  和  $\phi_i'^k$  收敛。当向用户推荐条目时, 模型会根据用户的历史交互数据进行本地更新, 生成个性化的推荐算法, 计算用户对所有尚未消费的条目的偏好程度, 再根据计算结果向用户推荐其最有可能感兴趣的条目。

#### 算法 1 基于 VC-BML 的用户偏好估计模型

输入: 任务分布  $p(\tau)$  和数据分布  $p(D|\tau)$ , 最大混合分量数  $K$ , 局部更新次数  $J$ , 局部更新的学习率  $\epsilon$ , 全局更新的学习率  $\zeta$

1. 初始化:  $\mu_1^k, \sigma_1^k; \forall k=1, \dots, K;$
2.  $\mu_1'^k, \sigma_1'^k, \forall k=1, \dots, K;$
3. 当不收敛进行
4.   取样一组用户  $B \sim p(\mathcal{B})$ ;
5.   初始化  $\eta_{n,k}, \forall k=1, \dots, K;$
6.   更新  $\gamma_k, \forall k=1, \dots, K-1;$
7.   //初始化式(1)中变分参数  $\psi_i^k = \{\mathbf{m}_i^k, \mathbf{\Lambda}_i^k\}$ ;
8.    $\mathbf{m}_i^k \leftarrow \mu_i^k, \forall k=1, \dots, K;$
9.    $\mathbf{\Lambda}_i^k \leftarrow \sigma_i^k, \forall k=1, \dots, K;$

10. 对于每个 B 中的用户  $t$  进行
11. 取样任务  $\tau_t \sim p(\tau)$ ;
12. 取样集群  $D_t = D_t^S, D_t^Q \sim p(D_t | \tau_t)$
13. 初始化  $\eta_{i,k}^t, \forall k=1, \dots, K$ ;
14. 更新  $\gamma_k^t, \forall k=1, \dots, K-1$ ;
15. //初始化输入层变分参数  $\Psi_t^k = \{\mathbf{m}_t^k, \Lambda_t^k\}$ ;
16.  $\mathbf{m}_t^k \leftarrow \mu_t^k, \forall k=1, \dots, K$ ;
17.  $\Lambda_t^k \leftarrow \sigma_t^k, \forall k=1, \dots, K$ ;
18. 当不收敛进行
19. 对于  $k=1$  转到  $K$  更新  $\eta_{i,k}^t$
20. 对于每个  $t=1, \dots, K$  进行
21. 从  $q(\theta_{i_t}^{\tau_t=k} | \mathcal{D}_t; \Psi_t, \Psi_t')$  里取样  $\theta_{i_t}^{\tau_t=k}$ ;
22. 更新  $\lambda_t^k \leftarrow \text{SGD}_J(\theta_{i_t}^{\tau_t=k}, \mathcal{D}_t^S, \epsilon)$
23. 更新  $\Psi_t^k \leftarrow \Psi_t^k + \zeta \nabla_{\Psi_t^k} \mathcal{L}(\lambda_t, \Psi_t, \eta_t, \gamma, \lambda_t', \Psi_t', \eta_t', \gamma')$ ;
24.  $\mu_{t+1}^k \leftarrow \mathbf{m}_t^k, \forall k=1, \dots, K$ ;
25.  $\Sigma_{t+1}^k \leftarrow \Lambda_t^k, \forall k=1, \dots, K$ ;
26. 当不收敛进行
27. 对于  $k=1$  转到  $K$  更新  $\eta_{i,k}^t$
28. 对于每个  $t=1, \dots, K$  进行
29. 从  $q(\theta_{i_t}^{\tau_t=k} | \mathcal{D}_t; \Psi_t, \Psi_t')$  里取样  $\theta_{i_t}^{\tau_t=k}$ ;
30. 更新  $\lambda_t \leftarrow \text{SGD}_J(\theta_{i_t}^{\tau_t=k}, \mathcal{D}_t^S, \epsilon)$ ;
31. 更新  $\Psi_t^k \leftarrow \Psi_t^k + \zeta \nabla_{\Psi_t^k} \mathcal{L}(\lambda_t, \Psi_t, \eta_t, \gamma, \lambda_t', \Psi_t', \eta_t', \gamma')$ ;
32.  $\mu_{t+1}^k \leftarrow \mathbf{m}_t^k, \forall k=1, \dots, K$ ;
33.  $\Sigma_{t+1}^k \leftarrow \Lambda_t^k, \forall k=1, \dots, K$ .

## 4 实验

### 4.1 用户偏好估计模型

本节通过实验对比和分析来验证基于 VC-BML 的推荐算法在服从非平稳分布的不同数据集上的表现。

将 VC-BML 与 5 种最先进的基于元学习和持续学习的推荐算法的模型进行了比较。

(1) Train-On-Everything (TOE)。这是一个很直观的对照组,每一次接收到任务  $t$  时其元参数都会被重新初始化,并在  $D_{1,t}$  上进行元训练。

(2) FTML<sup>[16]</sup>。利用 Follow the Leader 算法<sup>[17]</sup>来最小化元学习的损失函数。

(3) DPMM<sup>[18]</sup>。使用混合 Dirichlet 过程来模拟潜在的任务分布。

(4) BOMVI<sup>[19]</sup>。使用贝叶斯方法进行元学习和持续学习的变分推理。

(5) MeLU<sup>[5]</sup>。一个基于 MAML 的个性化用户偏好估计模型。

在 4 个数据集上进行实验: MovieLens 1M, Bookcrossing, Amazon music 和 Yelp restaurant。这几个数据集都提供了基本的用户和条目信息,如表 1 所列。

表 1 4 个数据集的基本信息

Table 1 Basic information of four datasets

	MovieLens	Bookcrossing	Amazon music	Yelp restaurant
number of users	6040	278 858	736 441	1 987 897
number of items	3 706	271 379	465 392	150 346
ratetimes	1 000 209	1 149 780	1 584 082	6 990 280
user information	gender, age, occupation, zip code	age	gender, age	sign up time, friends, comments
item information	year, genre, director, actor	year, author, publisher	year, genre, singer	address, category, opening hours
score range	1~5	1~10	1~5	1~5

将所提模型在以下结构下进行了测试。决策层设置为两层,每层有 64 个节点,所有嵌入向量的维度被设置为 32。局部更新的次数从 1 次到 5 次不等。算法 1 的批处理规模和最大迭代次数分别设置为 32 和 30。

本研究中使用了常用的平均绝对误差 (MAE) 和归一化的衡量搜索引擎质量指标 (nDCG) 来衡量推荐算法的性能。

$$MAE = \frac{1}{|U|} \sum_{i \in U} \frac{1}{|H_i^r|} \sum_{j \in H_i^r} |y_{ij} - \hat{y}_{ij}|$$

$$nDCG_k = \frac{1}{|U|} \sum_{i \in U} \frac{DCG_k^i}{IDCG_k^i}$$

$$DCG_k^i = \sum_{r=1}^k \frac{2^{R_r} - 1}{\log_2(1+r)}$$

其中,  $R_r$ ,  $U$  和  $IDCG_k^i$  分别是用户  $i$  对条目  $r$  的真实评分、测试用户集合和对于用户  $i$  最有可能的  $DCG_k^i$ 。

### 4.2 在服从非平稳分布不同数据集上的表现

实验中使用 4 个数据集,由于不同的数据集对应着不同的任务,因此从给定的数据集按顺序取样得到的任务遵循非平稳分布。依次在 3 个数据集上进行训练,首先需要选定一个数据集,然后从选定的数据集中随机抽出一连串的任务,再对被抽取的任务进行元模型训练。也就是说,首先对从 MovieLens 1M 数据集中抽取的任务进行元训练,多次抽取与训练过程完成后进入下一个数据集,依次在 3 个数据集上完成训练后,再在元测试集上对性能进行评估。在此过程中,我们根据验证集调整超参数。

表 2 展示了在每个元训练阶段,所提模型在已经学习过的数据集(即  $D_{1,t}$ )上的平均绝对误差。

表 2 在已经学过的数据集上的表现的平均值

Table 2 Average of performance over the data set that has been learned

Algorithm	MovieLens		Bookcrossing		Amazon music		Yelp restaurant	
	MAE	nDCG	MAE	nDCG	MAE	nDCG	MAE	nDCG
TOE	0.737	0.909	1.168	0.793	1.358	0.738	1.471	0.724
FTML	0.696	0.886	1.042	0.802	1.204	0.746	1.477	0.721
DPMM	<b>0.685</b>	<b>0.936</b>	1.012	0.809	1.172	0.771	1.214	0.745
BOMVI	0.773	0.887	1.021	0.788	1.124	0.769	1.115	0.766
MeLU	0.768	0.891	1.014	0.811	1.157	0.753	1.168	0.759
VCBML	0.724	0.897	<b>0.980</b>	<b>0.829</b>	<b>1.071</b>	<b>0.781</b>	<b>1.063</b>	<b>0.786</b>

当在不同的数据集上依次进行元训练时,基于 VC-BML 的推荐算法都取得了最佳性能,这表明 VC-BML 可以更好地处理服从非平稳分布的任务。我们认为 VC-BML 在 MovieLens 1M 上没有表现出最佳性能的原因,是 ELBO 中的 KL 散度引起的过度正则化。

为了更深入地讨论模型性能的变化情况,测试了所提模型在不同元训练阶段在 4 个数据集上的平均绝对误差的演变,结果如表 3 所列。尽管 FTML, DPMM 和 MeLU 在单一的任务上都取得了良好的性能(即同一数据集),但当任务分布转移到新的数据集时,它们的性能就会下降。相比之下,在不同的数据集上,VC-BML 在不同元训练阶段的元测试平均绝对误差是较为稳定一致的。这表明 VC-BML 有效地抵抗了负知识转移问题。

为了更深入解释所提模型在解决灾难性遗忘问题方面的性能,对比表 3 中元训练阶段开始(MovieLens 1M)和结束(Yelp restaurant)时对每个数据集进行元测试时的平均绝对误差。结果表明,参数算法(FTML, DPMM 和 MeLU)比贝叶斯算法(BOMVI 和 VC-BML)的遗忘问题更严重。例如,

在 MovieLens 1M 数据集上,参数算法的 MAE 从 0.696~0.737 上升到 0.835~0.938,而贝叶斯算法的 MAE 几乎没有变化(从 0.724~0.728 到 0.793~0.807)。这是因为参数算法允许模型快速大幅度地改变参数以快速适应新的任务,而由此会忘记以前的知识;而贝叶斯算法中的 KL 项作为一个正则项,可以抵抗这种遗忘。此外,当只在 MovieLens 1M 上进行元训练时,尽管贝叶斯算法在 MovieLens 1M 上的性能不是最好的,但它们在未见过的数据集(即 Bookcrossing, Amazon music 和 Yelp restaurant)上的性能总是优于参数化算法。这是因为参数算法是为特定的任务分布而定制的;而贝叶斯算法在对未知分布的任务进行学习时有更强的概括性,能够更好地适用于新任务。因此,贝叶斯算法更适合用于处理服从非平稳分布的任务。而我们所使用的基于 VC-BML 的推荐算法优于同为贝叶斯算法的 BOMVI 的原因是, BOMVI 模型只有一个元参数的分布,因此在处理不同的任务分布时受到了负知识转移的影响。

需要注意的是实验结果与元训练的顺序无关,我们可以通过调整不同数据集的顺序进行进一步的实验验证。

表 3 各阶段在各个数据集上测试时的平均绝对误差

Table 3 Mean absolute error of tests at each stage on each data set

Meta-training Stage	Algorithm	MovieLens	Bookcrossing	Amazon music	Yelp restaurant
MovieLens	TOE	0.737	2.864	1.642	3.211
	FTML	0.696	2.645	1.557	3.009
	DPMM	<b>0.685</b>	2.537	1.409	3.132
	BOMVI	0.773	2.332	1.274	2.973
	MeLU	0.768	2.418	1.384	3.02
	VC-BML	0.724	<b>2.247</b>	<b>1.295</b>	<b>2.896</b>
Bookcrossing	TOE	0.782	2.003	1.729	3.113
	FTML	0.783	1.885	1.623	2.997
	DPMM	0.748	1.738	1.398	3.131
	BOMVI	0.786	1.339	<b>1.266</b>	2.869
	MeLU	0.791	1.963	1.407	3.132
	VC-BML	<b>0.764</b>	<b>1.294</b>	1.286	<b>2.902</b>
Amazon music	TOE	0.805	2.419	1.118	3.132
	FTML	0.862	2.132	1.031	3.014
	DPMM	0.811	1.841	1.012	3.157
	BOMVI	0.792	2.332	1.006	2.935
	MeLU	0.803	2.418	1.095	3.221
	VC-BML	<b>0.778</b>	<b>1.317</b>	<b>0.992</b>	<b>2.881</b>
Yelp restaurant	TOE	0.824	2.723	1.272	2.773
	FTML	0.938	2.419	1.238	2.697
	DPMM	0.872	2.012	1.213	2.431
	BOMVI	0.807	1.553	1.031	1.824
	MeLU	0.835	2.116	1.177	1.994
	VC-BML	<b>0.793</b>	<b>1.332</b>	<b>0.978</b>	<b>1.635</b>

#### 4.3 超参影响:混合分量总数

为了提高计算效率,采用了 CRP 的有限截尾结构,对混合分量的数量设定了一个上限,即  $K$ 。为了研究  $K$  的影响,通过改变  $K$  来进行实验。表 4 列出了在每个元训练阶段,所提模型在已经学习过的数据集(即  $D_{1:t}$ )上的平均绝对误差。结果显示,当  $K=1$  时,VC-BML 的性能最差,这表明一个单一的元参数分布在面对不同任务时是不足够的。VC-BML 的性能随着  $K$  的增大而提升,这表明实例化更多的元参数分布有利于提升模型的性能。而当  $K>4$ (任务分布的实际数量)时,再增大  $K$ ,VC-BML 性能改进就不明显了。

表 4 各元学习阶段在已经学习过的数据集上的 MAE

Table 4 MAE of each meta-learning stage model on the learned

	dataset			
	MovieLens	Bookcrossing	Amazon music	Yelp restaurant
$K=1$	0.737	1.013	1.231	1.186
$K=2$	0.742	0.995	1.103	1.092
$K=4$	0.718	0.984	1.078	1.074
$K=6$	0.724	0.980	1.071	1.063

为了验证 VC-BML 能否识别不同的任务分布,我们在上述的 4 个数据集上用  $K=6$  对模型进行元训练,然后将  $\approx$ (潜在的任务类别)的后验可视化。具体来说,从每个数据集

的元测试集中随机抽出 100 个任务,并计算其分组的后验,结果如图 3 所示。该结果表明,基于 VC-BML 的推荐系统可以识别具有独立混合分量的不同任务分布。例如,分量 1 对应的是 MovieLens 1M,而分量 4 对应的是 Yelp restaurant。这有助于解释为何基于 VC-BML 的推荐系统在处理服从非平稳分布任务时具有优越的性能,因为它可以在不同的混合分量中存储不同任务分布的知识,并在类似任务出现时调用这些知识。此外,分量 5 和 6 的低后验值表明,VC-BML 只有在发现不同的任务时才会创建新的元参数分布。

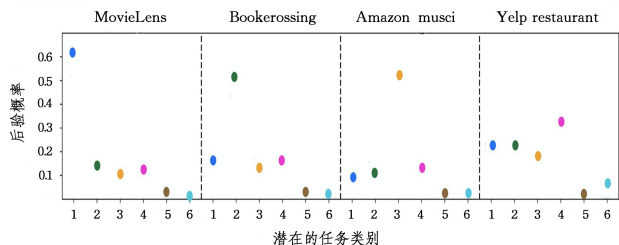


图 3 基于 VC-BML 的推荐算法的更新过程

Fig. 3 Update process of VC-BML based recommendation system

**结束语** 本文讨论了如何处理服从非平稳分布的推荐任务。为了抵抗负知识转移问题以及灾难性遗忘问题,构建一个能够适用于多种任务的推荐算法,我们提出了一种基于 VC-BML 的推荐算法,其中的元参数服从动态混合高斯模型。我们采用了中国餐馆过程来灵活地确定任务集群的数量,使用结构化变分推理方法来近似后验概率。大量的实验结果表明,在完成各种不同条目内容推荐时,相较于其他基准算法,基于 VC-BML 的推荐算法在性能上表现出了明显的优势。本文使用到的数据主要为用户或商品的基础信息,还有大量的文本、图像等上下文信息未被探索,作为未来进一步研究的方向,我将利用这些上下文来提升模型适应不同数据集的能力。

## 参考文献

- [1] MCNEE S M, RIEDL J, KONSTAN J A. Being accurate is not enough; how accuracy metrics have hurt recommender systems [C]//Extended Abstracts(CHI'06). 2006:1097-1101.
- [2] LI X, SHE J. Collaborative variational autoencoder for recommender systems[C]//KDD. 2017:305-314.
- [3] OORD A V D, DIELEMAN S, SCHRAUWEN B. Deep content-based music recommendation[C]//NIPS. 2013:2643-2651.
- [4] WEI J, HE J H, CHEN K, et al. Collaborative filtering and deep learning based recommendation system for cold start items[J]. Expert Systems with Applications, 2017, 69: 29-39.
- [5] LEE H, IM J, JANG S, et al. Melu: Meta-learned user preference estimator for cold-start recommendation[C]//KDD. 2019:1073-1082.
- [6] DENEVI G, STAMOS D, CILIBERTO C, et al. Online-within-online meta-learning[C]//NIPS. 2019:1-11.
- [7] VILALTA R, DRISSE Y. A perspective view and survey of meta-learning[J]. Artificial Intelligence Review, 2002, 18(2): 77-95.
- [8] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//ICML. 2017:1126-1135.
- [9] LIANG S S, YILMAZ E, KANOULAS E. Collaboratively tracking interests for user clustering in streams of short texts[J]. IEEE Transactions on Knowledge Data Engineering, 2018, 31(2):257-272.
- [10] ZHANG Q, FANG J Y, MENG Z Q, et al. Variational Continual Bayesian Meta-Learning[C]//NIPS. 2021:24556-24568.
- [11] CHENG H T, KOC L, HARMSEN J, et al. Wide deep learning for recommender systems[C]//DLRS. 2016:7-10.
- [12] HE X N, LIAO L Z, ZHANG H W, et al. Neural collaborative filtering[C]//WWW. 2017:173-182.
- [13] BARKAN O, KOENIGSTEIN N. Item2vec: Neural item embedding for collaborative filtering[C]//IEEE MLSP. 2016:1-6.
- [14] ZHANG F Z, YUAN N J, LIAN D F, et al. Collaborative knowledge base embedding for recommender systems [C]//KDD. 2016:353-362.
- [15] NAIR V, EHINTON G. Rectified linear units improve restricted boltzmann machines vinodnair[C]//ICML-10. 2010:807-814.
- [16] FINN C, RAJESWARAN A, KAKADE S, et al. Online meta-learning[EB/OL]. (2019-02-22) [2019-07-03]. <https://doi.org/10.48550/arXiv.1902.08438>.
- [17] KALAI A, VEMPALA S. Efficient algorithms for online decision problems[J]. Journal of Computer and System Sciences, 2005, 71(3):291-307.
- [18] JERFEL G, GRANT E, GRIFFITHS T, et al. Reconciling meta-learning and continual learning with online mixtures of tasks [EB/OL]. (2018-12-14) [2019-06-19]. <https://doi.org/10.48550/arXiv.1812.06080>.
- [19] YAP P, RITTER H, BARBER D. Bayesian online meta-learning with Online Laplace Approximation [J]. arXiv: 2005.00146, 2020.



**ZHU Wentao**, born in 2000, postgraduate. Her main research interests include recommendation system and meta-learning.



**LIU Wei**, born in 1989, Ph.D, associate researcher, is a member of China Computer Federation. His main research interests include recommendation system and spatio-temporal data mining.