

文章编号: 1000-5641(2017)05-0138-16

## 基于隐变量模型的多维用户偏好建模

王珊蕾<sup>1</sup>, 岳 昆<sup>1</sup>, 武 浩<sup>1</sup>, 田凯琳<sup>2</sup>

(1. 云南大学 信息学院, 昆明 650500; 2. 西南林业大学 图书馆, 昆明, 650224)

**摘要:** 从用户行为数据构建用户偏好模型, 是解决个性化服务、评分预测和用户行为定向等问题的重要基础. 本文从用户的评分数据出发, 以多个隐变量分别描述用户在评分对象多个维度的偏好, 以含有多个隐变量的贝叶斯网 (简称隐变量模型) 作为表示用户偏好的基本知识框架. 首先根据用户偏好和隐变量的特定含义给出模型构建的约束条件, 进而提出基于约束条件的模型构建方法, 使用约束条件下的EM算法来计算模型参数, 约束条件下的SEM算法来构建模型结构. 针对多隐变量情形下模型构建过程中产生大量中间数据带来的计算复杂度急剧上升的问题, 本文使用Spark计算框架实现模型构建的方法. 建立在Movielens数据集上的实验表明, 本文提出的方法是有效的.

**关键词:** 评分数据; 多维偏好; 隐变量; 贝叶斯网; Spark

**中图分类号:** TP311 **文献标志码:** A **DOI:** 10.3969/j.issn.1000-5641.2017.05.013

## Modeling multi-dimensional user preference based on the latent variable model

WANG Shan-lei<sup>1</sup>, YUE Kun<sup>1</sup>, WU Hao<sup>1</sup>, TIAN Kai-lin<sup>2</sup>

(1. *School of Information Science and Engineering, Yunnan University, Kunming 650500, China;*  
2. *Library of South West Forestry University, Kunming 650224, China*)

**Abstract:** Modeling user preference from user behavior data is the basis of personalization service, score prediction, user behavior targeting, etc. In this paper, multi-dimensional preferences from rating data are described by multiple latent variables and the Bayesian network with multiple latent variables is adopted as the preliminary knowledge framework of user preference. Constraint conditions are given according to the inference of user preference and latent variables, upon which we propose a method for modeling user preference. Parameters are computed by EM algorithm and structure is established by SEM algorithm with respect to the given constraints. In the case of multiple latent

收稿日期: 2017-05-01

基金项目: 国家自然科学基金(61472345, 61562090); 教育部“云数融合、科教创新”基金(2017B00016); 第二批“云岭学者”培养项目(C6153001); 云南省应用基础研究计划重点项目(2014FA023); 云南大学青年英才培育计划(WX173602); 中国博士后科研基金(2016M592721)

第一作者: 王珊蕾, 男, 硕士研究生, 研究方向为海量数据分析与知识发现. E-mail: 407773704@qq.com.

通信作者: 岳 昆, 男, 博士, 教授, 博士生导师, 研究方向为海量数据分析与知识发现.

E-mail: kyue@ynu.edu.cn.

variables, a large amount of intermediate data is generated in modeling, which causes the increasing computational complexity. Therefore, we implement the modeling method with Spark computing framework. Experiments results on the Movielens dataset verify that the method proposed in this paper is effective.

**Key words:** rating data; multi-dimensional preference; latent variable; Bayesian network; Spark

## 0 引言

随着移动互联网和 Web2.0 的快速发展, 互联网已经渗入到人们生活工作的方方面面, 随之产生了大量的用户行为数据. 例如, 用户的位置信息数据、用户对电影或音乐的评分数据、电子商务应用中用户对商品做出评价而产生的用户评分数据, 这些数据的不断产生使得用户行为建模成为可能. 用户行为建模对个性化信息服务、行为定向等问题的解决有重要的作用; 分析用户行为数据, 建立用户偏好模型是实现和提供这些服务的基础和关键, 具有重要意义.

一般的用户评分数据包含用户属性信息、评分对象属性信息以及用户评分. 用户偏好是用户对事物的喜好或倾向性的选择, 不能被直接观测到. 用户评分数据反映了用户偏好, 用户偏好也从一定程度上决定了用户的评分. 实际中, 用户对事物的喜好或倾向选择会受到事物本身所固有的多个属性的影响. 例如, MovieLens 数据集<sup>[1]</sup>包括用户信息、电影信息和评分, 电影信息包括多个维度的电影属性, 如年代、类型、语言等; 而用户对每个维度的电影属性都会有相应的喜好或倾向, 形成了多个维度的用户偏好. 这意味着, 用户对电影最终的倾向会受多维偏好的影响, 对多维偏好的分析是获得精确用户倾向或喜好的必要条件. 同时, 多个维度的用户偏好之间也可能相互影响, 这使得准确描述多维偏好、建立用户偏好模型具有实际意义, 也具有一定的挑战.

近年来, 关于多维用户偏好建模已经有了一些研究, 使用向量模型或主题模型等表达多维用户偏好<sup>[2-6]</sup>. 例如, LDA (Latent Dirichlet Allocation)<sup>[4]</sup>是一种具有代表性的主题模型, SVD (Singular Value Decomposition)<sup>[6]</sup>是协同过滤中常见的评分预测算法, 这些模型能够表达预先给定的依赖关系. 但是, 评分数据中各个属性间存在相互影响的依赖关系, 且具有不确定性, 例如, 评分、用户属性以及电影属性之间会相互影响, 且影响的方向以及程度不确定. 这些模型能够表达预先给定的依赖关系, 但评分数据中任意的、不确定性的依赖关系的表示, 还需进一步探索.

贝叶斯网 (BN, Bayesian Network)<sup>[7]</sup>是由一组节点构成的有向无环图 (DAG, Directed Acyclic Graph), 其中每个节点都有一个条件概率表 (CPT, Conditional Probability Table). BN 是表达属性间任意的依赖关系以及不确定性的有效工具<sup>[8]</sup>, 且具有优秀的推理能力. 使用隐变量描述用户偏好, 将隐变量引入 BN 构成含隐变量的 BN, 简称为隐变量模型 (LVM, Latent Variable Model), 是本文研究的基本思想. 为了客观有效地描述评分数据中任意的、不确定的依赖关系, 我们以多个隐变量分别描述多个维度的用户偏好, 以隐变量模型作为表示各变量之间依赖关系的基本知识框架, 进而从用户评分数据构建隐变量模型.

隐变量无法被直接观测到<sup>[9]</sup>, 参数的计算需要先对隐变量进行填充, 不能直接使用最大似然估计来计算参数. 期望最大算法 (EM, Expectation Maximization)<sup>[10]</sup>是一种可以对隐变量进行填充并寻找参数最大似然或最大后验概率的有效算法. 结构期望最大算法 (SEM,

Structure Expectation Maximization)<sup>[11]</sup>是一种结合了EM算法的结构打分搜索方法,能够在EM迭代中直接优化打分,可以有效的构建隐变量模型的结构.基于此,本文基于EM算法提出隐变量模型参数的计算方法,基于SEM算法提出结构的构建方法.

一方面,SEM算法和EM算法的运行结果均会受初始结构和约束条件的影响<sup>[12]</sup>.Friedman等<sup>[11]</sup>指出,随机的初始结构会导致SEM算法很难得到一个有意义的运行结果.Jin等<sup>[13]</sup>也已经证明,随机初始参数也会导致EM算法容易收敛到局部极值点.基于此,为了保证模型构建的有效性,本文对EM算法和SEM算法进行了扩展,从初始值的约束限定出发,提出了基于约束条件的模型构建方法.另一方面,EM算法和SEM算法的执行,都涉及大量的迭代计算,而每一次迭代计算又涉及NP困难的概率计算,计算复杂度较高.Spark是一种基于内存的并行计算框架,所有中间结果暂存在内存中,可以处理高复杂度的问题,而且擅长迭代计算.基于此,本文使用Spark计算框架实现基于约束的EM算法以及基于约束的SEM算法的并行执行.

总的来说,本文的主要内容可以概括如下:

- 从用户评分数据出发,用多个隐变量分别表示多个维度的用户偏好,以含多隐变量的贝叶斯网(隐变量模型)来构建多维偏好模型.
- 提出约束条件下的模型构建方法,包括约束下的参数计算方法以及约束下的结构构建方法,并利用Spark实现模型构建方法.
- 建立在MovieLens数据集上的实验,验证了本文方法的高效性和有效性.

## 1 相关工作

从评分数据出发构建偏好模型方面,Zhao等<sup>[14]</sup>提出评价数据较少情形下的商品服务评估模型.文献[15]基于上下文感知的方法来获取用户偏好.文献[16]以条件偏好网络模型来构建用户偏好.文献[17]基于上下文最小二乘支持向量机来构建用户偏好.在多维偏好建模方面,也出现了大量研究.Kassak<sup>[2]</sup>等以对象的多个属性特征来描述对象,以向量描述多维偏好.Zhao等<sup>[5]</sup>融合了类型主题模型和地域主题模型,构建了一种二维偏好模型(对地域的偏好和对类型的偏好).这些方法只能表达预先给定的相互影响的依赖关系,而数据中任意形式的依赖关系的表示,还需进一步研究.

基于BN或隐变量模型的单维用户偏好建模方面,Kim等<sup>[9]</sup>用隐变量表示用户的评价行为,用含隐变量的BN来构建偏好模型.Gao等<sup>[18]</sup>用隐变量描述用户对电影类型的偏好,进而构建偏好模型.Huang等<sup>[19]</sup>从旅游的领域知识构建BN,在此基础上估计用户的旅游偏好.Chapelle等<sup>[20]</sup>使用隐变量刻画用户兴趣,预定义模型的结构,提出了用以描述用户点击行为的动态贝叶斯网模型.这些方法为我们的研究提供了参考,但是以隐变量模型为基本框架来构建多维偏好模型的方法还需进一步研究.

在利用BN进行多维偏好建模方面,Huete等<sup>[21]</sup>用BN表示用户画像,将用户对商品多个维度的评分视为隐变量,根据各维度评分与最终评分之间的特定关系构建BN结构,进而根据BN的推理来预测商品的评分.Auffenberg等<sup>[22]</sup>用多个隐变量来刻画用户应对温度变化可能的多个类型的倾向选择,以给定的依赖关系来构建BN结构.上述基于隐变量模型的偏好建模方法,以隐变量表示用户偏好,以含隐变量的BN来构建偏好模型,这些方法为我们的研究提供了参考,但通常是先给定BN的图结构.以隐变量模型为基本框架,从评分数据构建能客观表达不确定性的多维偏好模型,还需进一步研究.

基于数据分析的隐变量模型构建方面, 我们<sup>[23]</sup>基于 MapReduce 模型, 从打分搜索的方法入手, 以 MDL 的数值来打分选择贝叶斯网模型. Yoshinori 等<sup>[24]</sup>对搜索空间进行了分布式的处理, 基于动态规划的思想提出了一种最优结构的搜索算法. 这些隐变量模型构建方法为本文的研究提供了参考, 但针对含多个隐变量的 BN 构建方法还需进一步探索.

## 2 相关定义

$R = \{r_1, r_2, \dots, r_k\}$  是用户对评分对象的评分集合,  $S = \{S_1, S_2, \dots, S_m\}$  为用户属性的集合,  $I = \{I_1, I_2, \dots, I_n\}$  为评分对象属性的集合, 隐变量的集合  $L$  表示用户的  $n$  维偏好,  $L = \{L_1, L_2, \dots, L_n\}$ .  $I_i$  为评分对象第  $i$  个属性的取值.  $L_i$  表示用户对第  $i$  个属性的偏好 ( $1 \leq i \leq n$ ), 即用户对评分对象第  $i$  个属性所倾向的取值.  $L_i, I_i \in \{a_{i1}, a_{i2}, \dots, a_{il_i}\}$ ,  $l_i$  为第  $i$  个属性的可能取值个数.

例如, 电影评分数据中,  $R = \{1, 2, 3, 4, 5\}$ ,  $S_1 \in \{\text{男}, \text{女}\}$ ;  $S_2 \in \{\text{律师}, \text{医生}, \text{教师}\}$ ;  $I_1, L_1 \in \{\text{动作}, \text{喜剧}\}$ ;  $I_2, L_2 \in \{80\text{年代}, 90\text{年代}\}$ ;  $I_3, L_3 \in \{\text{英语}, \text{汉语}\}$ . 对于一条评分数据, 用户属性描述为  $S = \{S_1(\text{性别}) = \text{女}, S_2(\text{职业}) = \text{律师}\}$ , 评分对象属性描述为  $I = \{I_1(\text{类型}) = \text{动作}, I_2(\text{年代}) = 90\text{年代}, I_3(\text{语言}) = \text{英语}\}$ , 用户对评分对象各属性的偏好描述为  $L = \{L_1(\text{类型}) = \text{动作}, L_2(\text{年代}) = 90\text{年代}, L_3(\text{语言}) = \text{汉语}\}$ .

下面给出多维偏好模型的形式化定义.

**定义 1** 一个多维用户偏好模型是一个含多个隐变量的贝叶斯网, 即多维隐变量模型 (MLVM, Multiple Latent Variables Model), 表示为二元组  $(\varphi, \theta)$ , 其中:

- $\varphi = (V, E)$  是模型的有向无环图结构, 其中,  $V$  为图中节点的集合,  $V = S \cup L \cup I \cup R$ .  $E$  是有向边的集合, 节点间的边表示的变量之间的直接依赖关系.
- $\theta$  是模型的参数 (即 CPT) 集合,  $\theta$  表示为联合概率  $P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \prod_{i=1}^n P(X_i = x_i | \pi(X_i))$ , 为所有节点的条件概率的乘积.

## 3 多维用户偏好模型的构建

### 3.1 约束条件

为了保证模型构建的有效性, 本文对 EM 算法和 SEM 的初始值进行约束限定. 本文从用户评分数据出发, 根据用户偏好和隐变量的特定含义给出模型构建的初始值需要满足的约束条件.

**约束 1** 第  $i$  维的偏好  $L$  表达用户对第  $i$  维属性  $I_i$  的倾向, 多维偏好和评分对象的多个属性一一对应; 用户属性  $\{S_1, S_2, \dots, S_m\}$  不依赖于其它变量. 如图 1 所示.

**约束 2** 用  $L_i = I_i$  描述用户的第  $i$  维偏好和对象的第  $i$  个属性取值相一致的情形.  $\{L_1, L_2, \dots, L_n\}$  与  $\{I_1, I_2, \dots, I_n\}$  各维度取值相一致的属性总数为  $n_1$ , 不一致的属性总数为  $n_2$ ,  $0 \leq n_1, n_2 \leq n$ ,  $n_1 + n_2 = n$ . 若  $n_1 > n_2$ , 则用户更可能倾向或喜好该对象会打高分, 反之打低分. 这一约束用公式 (1) 描述:

$$P(R = R_1 | L, I, n_1 > n_2) > P(R = R_1 | L, I, n_2 > n_1). \quad (1)$$

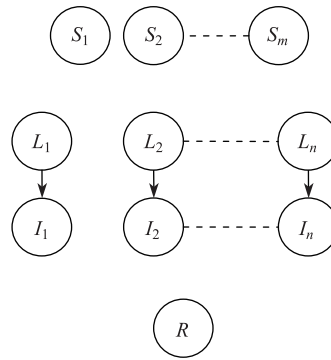


图1 结构约束

Fig. 1 Structural constraint

### 3.2 基于约束条件的参数计算

EM算法从一个随机初始参数开始,先对数据集中隐变量的值进行填充,然后计算并更新参数,迭代直至收敛.假设数据集 $D_t$ 中一次用户评分记录为一个样本,若已经进行了 $t$ 次迭代,第 $t+1$ 次迭代的执行过程由E-步和M-步完成.

#### E-步

首先,根据当前缺值数据集 $D_t$ 以及当前参数集 $\theta_t$ 填充数据集.根据公式(2)计算不同隐变量取值组合的概率,使用该取值及概率来填充数据.

$$P(L = j|D_t, \theta_t) = \frac{P(L = j, D_t|\theta_t)}{\sum_{j=1}^c P(L = j, D_t|\theta_t)}. \quad (2)$$

其中, $L$ 表示隐变量的集合, $j$ 表示隐变量取值的集合.

进而,根据公式(3)计算充分统计量 $m_{ijk}^t$ ,其中 $V_i$ 是第 $i$ 个节点, $\pi(V_i)$ 是节点 $V_i$ 的父节点集合.

$$m_{ijk}^t = \sum_{l=1}^m P(V_i = k, \pi(V_i) = j|D'_l). \quad (3)$$

#### M-步

然后,使用公式(4)计算本次迭代所求的参数值 $\theta_{t+1}$ .

$$\theta_{t+1} = \frac{m_{ijk}^t}{\sum_{k=1}^{r_i} m_{ijk}^t}. \quad (4)$$

为了确保收敛的效率,我们给定一个阈值来度量两次迭代所得参数的相似程度.如公式(5)所示, $\ln P(D_{t+1}|\theta_{t+1})$ 和 $\ln P(D_t|\theta_t)$ 分别是第 $t+1$ 次和第 $t$ 次迭代所得参数的对数似然函数,若 $S(\theta_t, \theta_{t+1}) < \partial$  ( $\partial$ 为参数相似度阈值),则认为参数已经收敛,迭代结束.

$$S(\theta_t, \theta_{t+1}) = \ln P(D_{t+1}|\theta_{t+1}) - \ln P(D_t|\theta_t). \quad (5)$$

EM算法迭代的每一步,都会对隐变量的值进行填充、并生成新的CPT.每个隐变量都有多个可能的填充值,对所有隐变量的值进行填充后,评分数据集集中的数据量随隐变量个数

的增加呈阶乘数量级增长. 同时, 隐变量 CPT 的规模也随隐变量父节点个数的增加呈阶乘数量级扩大. 数据填充时大量的中间结果及对其频繁的读写操作, 对数据存储及读写速度提出了新挑战, 传统的单任务计算框架已经无法满足需求.

因此, 本文基于 Spark 框架设计了并行的参数计算方法. 首先随机产生一组满足约束 2 的初始参数; 然后, 以这组参数作为 EM 算法的初始值, 执行以下步骤.

1) 数据集被分割成  $q$  个数据块, 并分配给  $q$  个 Map 函数. 针对每个数据块中的每个样本, 根据公式 (2) 进行样本数据填充, 然后收集每个 Map 函数的结果, 根据公式 (3) 计算充分统计量.

2) 根据公式 (4) 计算参数.

步骤 1) 和 2) 迭代执行, 直至收敛, 执行流程如图 2 所示, 上述思想见算法 1.

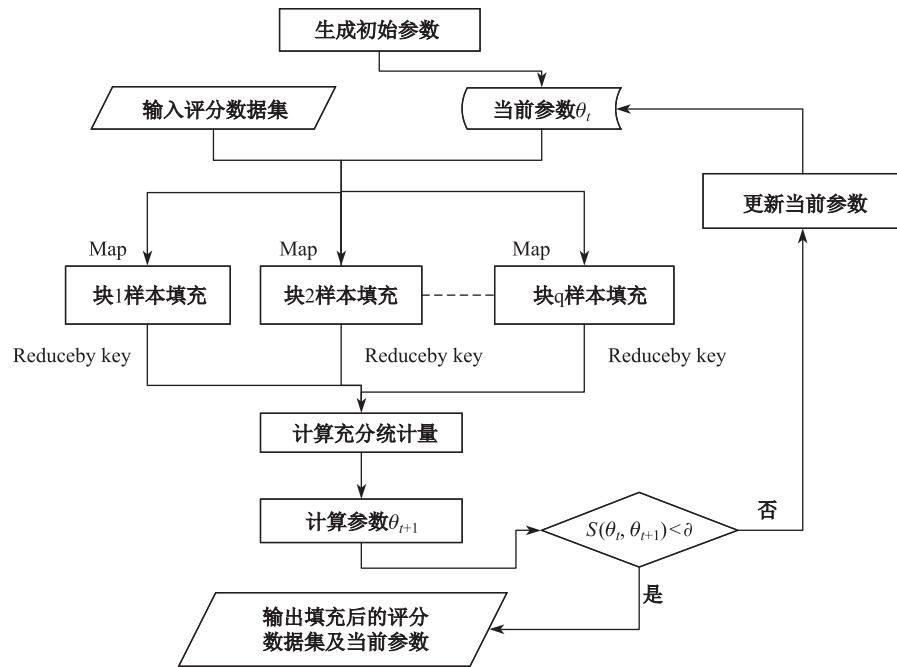


图2 算法1的流程

Fig. 2 Process of Algorithm 1

**算法 1:** CPT-Learn( $\varphi^0, D, \theta^0, \delta, T$ )

**输入:**  $\varphi^0$ : 当前模型结构

$D$ : 当前数据集

$\theta^0$ : 当前参数 (初次迭代则为空)

$\delta$ : 相似度阈值

$T$ : 迭代次数

**输出:**  $D'$ : 完整数据集

$\theta'$ : 模型参数

1:  $\varphi \leftarrow \varphi^0$  //初始化结构

2: **if**  $\theta^0$  为空 **then**

```

3: 随机产生一组满足约束 2 的初始参数  $\theta'$ 
4:  $\theta^0 \leftarrow \theta'$  //将初始参数作为当前参数
5: end if
6: for  $t \leftarrow 0$  to  $T$  do
7: pair  $\leftarrow D.map\{Line \Rightarrow$  //对每一行数据操作
    key  $\leftarrow$  填充该行数据的缺值
    value  $\leftarrow$  根据公式 (2) 计算当前填充组合的概率
    Emit(key, value)
    }
8: Rpair  $\leftarrow$  pair.flatMapValues{Line  $\Rightarrow$  //对pair中的每个键值对操作
    for  $i \leftarrow 0$  to key.length do //遍历 key 中的每一个值
        key  $\leftarrow (V_i = v_i, \pi(V_i) = j)$ 
        //  $V_i$  是 key 中第  $i$  个值  $v_i$  对应的节点编号,  $\pi(V_i)$  是节点  $V_i$  的父节点集合
        Emit(key, value)
    end for
    }
9:  $m_{ijk}^i \leftarrow$  Rpair.reduceByKey() //按 key 求和
10:  $\theta^{t+1} \leftarrow$  根据公式 (4) 计算参数
11: if  $S(\theta^t, \theta^{t+1}) < \partial$  then //根据公式 (5) 计算  $S$ 
12: return  $\theta^{t+1}$ 
13: end if
14: end for

```

例如, 表 1 给出了用户评分数据片段的示例,  $L_1$  和  $L_2$  分别有 0 和 1 两种取值可能. 以图 3(a) 为模型当前结构, 执行算法 1, 迭代一次得到的参数如图 4 所示, 对样本 ID 为 0 的样本进行隐变量值的填充, 结果如表 2 所示. 评分  $R$  依赖于  $L_1$ 、 $L_2$ 、 $I_1$  和  $I_2$ ,  $R$  的条件概率表如图 3(b) 所示.

表 1 数据集片段

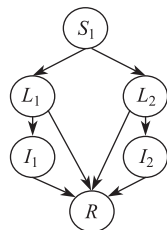
Tab. 1 Fragment of data set

样本 ID	$U_1$	$L_1$	$L_2$	$I_1$	$I_2$	$R$
0	0			0	1	1
1	1			0	1	0
...	...			...	...	...

表 2 填充后数据

Tab. 2 Filled data

样本 ID	$U_1$	$L_1$	$L_2$	$I_1$	$I_2$	$R$
0-1	0	0	0	0	0	1
0-2	0	1	0	0	1	1
0-3	0	0	1	0	1	1
0-4	0	1	1	0	1	1



$L_1, L_2, I_1, I_2$	$P(R=1)$	$P(R=2)$
0,0,0,0	0.8	0.2
0,0,0,1	0.7	0.3
...	...	...
1,1,1,1	0.4	0.6

图 3 当前结构和部分参数

Fig. 3 Current structure and partial parameters

$P(U_1)$			$P(L_1 U_1)$		
$P(U_1=1)$		$P(U_1=2)$	$U_1$	$P(L_1=1)$	$P(L_1=2)$
0.5		0.5	1	0.92	0.08
			2	0.81	0.19

$P(R L_1, L_2, I_1, I_2)$			$P(I_1 L_1)$		
$L_1, L_2, I_1, I_2$	$P(R=1)$	$P(R=2)$	$L_1$	$P(I_1=1)$	$P(I_1=2)$
0,0,0,0	0.80	0.20	1	0.88	0.12
0,0,0,1	0.44	0.56	2	0.34	0.66
.....	.....	.....			

图4 参数集

Fig. 4 Set of parameters

### 3.3 基于约束的结构构建

SEM算法是一种基于打分搜索的结构构建算法, 贝叶斯信息准则(BIC, Bayesian Information Criterion)是一种常用的有效打分标准, 能在缺值样本前提下对结构进行打分. 其计算公式如公式(6)所示.

$$\text{BIC}(\varphi|D) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} m_{ijk} \ln \frac{m_{ijk}}{m_{ij}} - \sum_{i=1}^n \frac{q_i(r_i - 1)}{2} \ln N. \quad (6)$$

其中,  $i$ 、 $j$  和  $k$  分别表示节点编号、 $i$  节点父节点的取值组合和  $i$  节点的取值,  $N$  表示样本个数,  $m_{ijk}$  表示充分统计量,  $m_{ij}$  表示充分统计量按  $k$  求和. SEM的基本思想是遍历每个节点, 对于每一个节点做加、减、转边的操作产生一系列候选模型. 接着, 根据公式(6)对候选模型进行BIC打分, 挑选分数最高的候选模型作为当前模型. 最后, 构建当前结构的参数. 迭代直至所有节点遍历结束.

本文基于Spark框架设计模型结构构建的并行算法. 首先从约束1出发, 随机产生一组满足约束1的初始结构, 然后随机生成一组满足约束2的初始参数. 以生成的初始参数和结构作为SEM算法的初始值, 对每个节点:

1) 调用算法1对数据集中的隐变量值进行填充, 并将充分统计量持久化(保存直至本次迭代结束), 以便计算BIC分数值. 收集完整数据集以及充分统计量, 通过加、减和转边生成一系列候选模型.

2) 为每个候选模型分配一个Map函数, 由步骤1)得到充分统计量计算候选模型的BIC分数; 收集每个Map函数的结果, 选择BIC分数最大的作为当前模型, 并调用算法1构建参数.

上述步骤1)和2)迭代执行, 直到所有节点遍历结束. 上述思想见算法2.

**算法2:** DAG-Learn( $\varphi^0, D, \theta^0, \partial, T$ )

**输入:**  $\varphi^0$ : 当前结构

$\theta^0$ : 当前参数

$D$ : 数据集 ( $|D| = N$ )

$\partial$ : 参数相似度阈值

$T$ : 迭代次数

$V\_num$ : 节点个数

**输出:**  $(\varphi, \theta)$ : 隐变量模型

1: **if**  $\varphi^0$  和  $\theta^0$  为空 **then**



```

2: 随机产生满足约束 1 的 BN 结构  $\varphi'$ ; 随机产生满足约束 2 的初始参数  $\theta'$ 
3:  $(\theta^0, \varphi^0) \leftarrow (\theta', \varphi')$  //将初始结构和参数作为当前结构和当前参数
4: end if
5:  $(\varphi, \theta) \leftarrow (\varphi', \theta')$ 
6:  $D^0 \leftarrow \text{CPT-Learn}(\varphi^0, D, \theta^0, \partial, 1)$  //调用算法 1 填充数据
7:  $\text{oldScore} \leftarrow \text{BICe}(\varphi, \theta | D^0)$  //计算当前 BIC 分数
8:  $\text{newScore} \leftarrow -\infty$ 
9: for  $i \leftarrow 0$  to  $V_{\text{num}} - 1$  do
10:  $c\_set \leftarrow$  对当前结点加、减或转边得到一系列候选结构
11:  $\text{Rpair} \leftarrow c\_set.\text{map}\{\text{Line} \Rightarrow$  //对  $c\_set$  中每一个候选结构进行操作
     $\theta^i, D^i \leftarrow \text{CPT-Learn}(\varphi^i, D^i, \theta^i, \partial, 1)$  //一次参数计算
     $\text{key} \leftarrow (\varphi^i, \theta^i)$ 
     $\text{value} \leftarrow \text{BICe}(\varphi^i, \theta^i | D^i)$  //计算候选结构 BIC 分数
     $\text{Emit}(\text{key}, \text{value})$ 
   $\}$ 
12:  $\text{Rpair}.\text{foreach}\{\text{Line} \Rightarrow$  //挑选 BIC 分数最大的候选模型
    if  $\text{value} > \text{newScore}$  then
       $(\varphi^i, \theta^i) \leftarrow \text{key}$ 
       $\text{newScore} \leftarrow \text{value}$ 
    end if
   $\}$ 
13: if  $\text{newScore} > \text{oldScore}$  then
14:  $(\theta^{i+1}, D^{i+1}) \leftarrow \text{CPT-Learn}(\varphi^i, D^i, \theta^i, \partial, T)$  //T 次参数计算
15:  $(\varphi, \theta) \leftarrow (\varphi^{i+1}, \theta^{i+1})$ 
16:  $\text{oldScore} \leftarrow \text{BICe}(\varphi, \theta | D^{i+1})$  //更新 BIC 分数
17: else
18: return  $(\varphi, \theta)$ 
19: end if
20: end for

```

例如, 图 5 为当前模型及参数, 执行算法 2, 对节点  $S_1$  操作, 通过加、减、转边得到一系列候选模型, 若图 6(a) 结构的 BIC 分值最大, 则选取为当前模型.

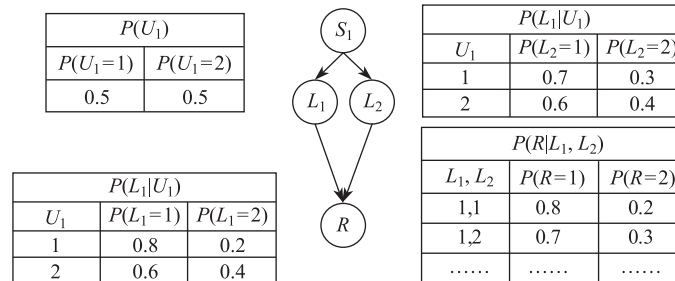


图 5 当前结构和参数

Fig. 5 Current structure and parameters

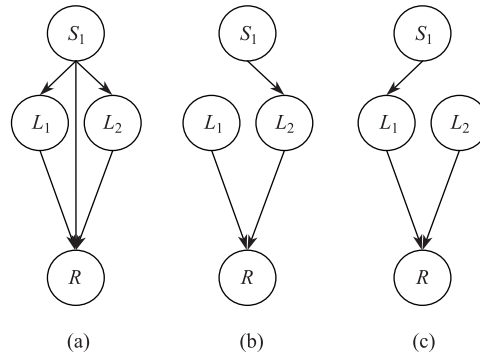


图6 候选结构

Fig. 6 Candidate structure

## 4 实验结果

本文使用 MovieLens 数据集<sup>[1]</sup>作为测试数据, 包括 3 952 条电影信息数据, 1 000 209 条评分数据以及 6 040 条用户信息. 预处理后的数据集共有 1 000 209 行, 每行数据是一次用户评分记录; 每行数据都有 2 个用户属性, 5 个电影属性以及 1 个评分值. 实验环境如下: 在两台主频 2.3 GHZ 的双路 HP 服务器上开辟了 7 台虚拟机, 1 台作为主节点、其余 6 台作为计算节点搭建 Spark 集群; 主节点分配 4 个 CPU 核心 16 GB 内存, 每个计算节点分配 8 个 CPU 核心 16 GB 内存, 将一个 CPU 计算核心简称为 C.

### 4.1 参数计算效率测试

我们首先测试了隐变量模型包括 7 个节点 (其中 2 个隐变量) 情况下, 不同计算核心个数、不同数据量情形下算法 1 的执行时间, 如图 7 所示. 可以看出, 算法 1 的执行时间随着数据量的增大而增加, 并且计算节点越多执行时间越少. 这说明算法 1 能够在较大数据量情形下能有效地学习隐变量模型的参数, 且具有较好的可扩展性.

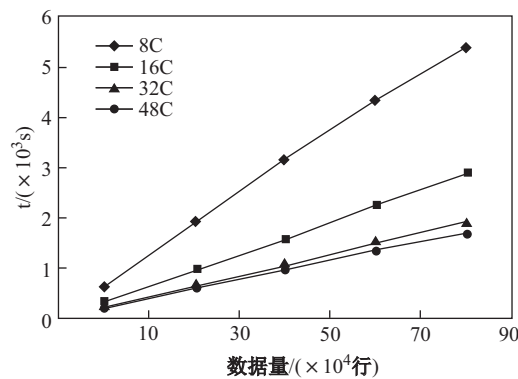


图7 算法1的执行时间

Fig. 7 Execution time of Algorithm 1

加速比是串行算法的执行时间与并行算法的执行时间的比值. 随着数据量的增加, 算法 1 在不同计算核心时的加速比如图 8 所示. 可以看出, 算法 1 的加速比随计算核心的增多而增大, 随数据量的增大而逐渐稳定, 这说明算法 1 有较好的可扩展性.

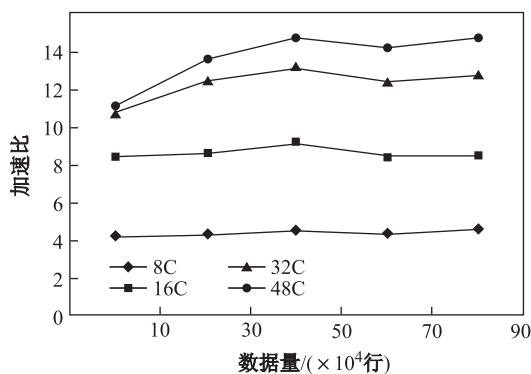


图8 算法1的加速比

Fig. 8 Speedup ratio of Algorithm 1

然后, 我们测试了10万行数据时, 隐变量个数对算法1执行时间的影响. 如图9所示, 算法1的执行时间随隐变量的个数增加呈阶乘数量级增加, 其原因在于隐变量的增加会使得填充后的数据量呈阶乘数量级增加, 算法1的执行时间也随之上升.

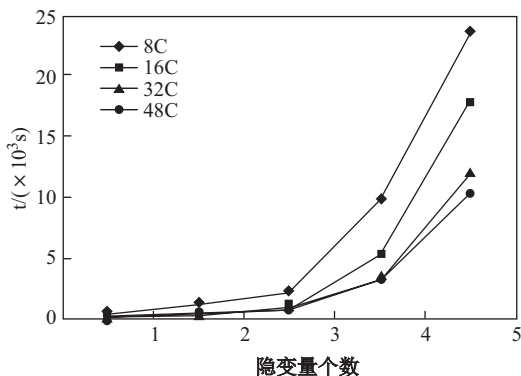


图9 隐变量个数增加时算法1的执行时间

Fig. 9 Execution time of Algorithm 1 with the increase of latent variables

接着, 我们测试了10万行数据、2个隐变量(与加速比实验保持一致)的情况下, 算法1的执行时间与隐变量父节点个数的关系, 如图10所示. 可以看出算法1的执行时间随隐变量父节点的增多而快速增大. 这说明, 随着隐变量父节点数的增加, 模型的依赖关系更加密集, 条件概率表的规模成阶乘数量级增大, 算法1的执行时间也随之上升, 隐变量父节点数是影响模型参数计算效率的瓶颈.

#### 4.2 结构构建效率测试

类似地, 本文对算法2进行了测试. 首先测试隐变量模型包括7个节点(其中2个隐变量)的情况下, 不同数据量情形下算法2的执行时间, 如图11所示. 可以看出, 算法2的执行时间随着数据量的增加而增加, 计算节点越多、执行时间越少, 这说明算法2在较大数据量情形下能有效地构建隐变量模型结构, 且具有较好可扩展性. 我们进一步测试了算法2的加速比, 如图12所示, 可以看出, 算法2的加速比随计算核心数的增加而增大, 说明算法2具有较好可扩展性.

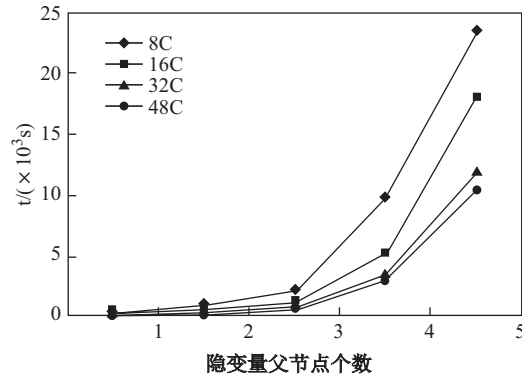


图 10 隐变量父节点数增加时算法 1 的执行时间

Fig. 10 Execution time of Algorithm 1 with the increase of parent nodes of the latent variable

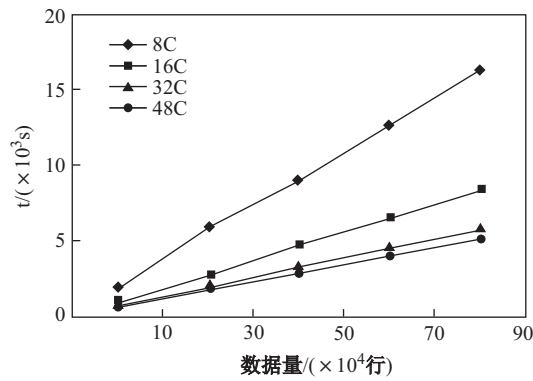


图 11 算法 2 的执行时间

Fig. 11 Execution time of Algorithm 2

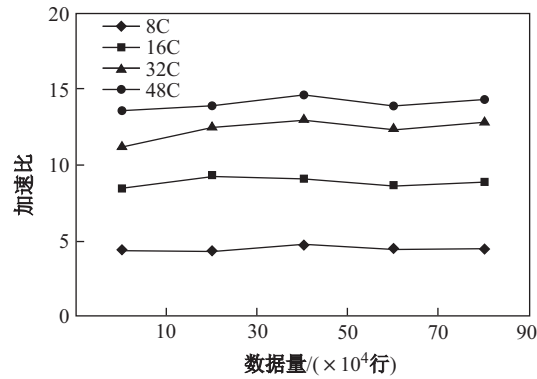


图 12 算法 2 的加速比

Fig. 12 Speedup ratio of Algorithm 2

然后, 我们对 90 万行数据, 不同计算核心, 隐变量模型包括 7 个节点 (其中 2 个隐变量) 的情况下, 算法 1 和算法 2 的执行时间进行了对比。如图 13 所示, 算法 2 的执行时间远高于算法 1, 随着计算核心的增多, 算法 1 和算法 2 的执行时间都有明显减少, 这从一定程度上说明本文方法具有较好的可扩展性。

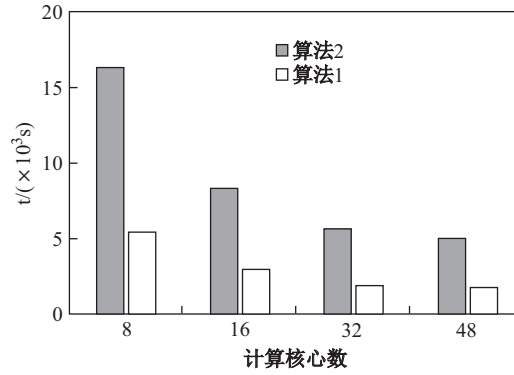


图 13 执行时间对比

Fig. 13 Comparison of execution time

#### 4.3 有效性测试

BN的推理是用BN的结构和参数来计算条件概率 $P(Q|E=e)$ , 其中 $Q$ 为查询变量,  $E$ 为证据变量. 变量消元法是一种有效的BN推理算法<sup>[6]</sup>, 适用于小规模BN的推理以及对推理精度要求较高的情况. BN工具箱(BNT)<sup>[25]</sup>是一个开源的Matlab软件包, 提供了以变量消元法为基础的推理工具, 设定模型结构和参数, 并确定查询变量和证据变量, 即可调用该推理工具进行推理. 为了测试模型的有效性, 我们从预处理过的Movielens数据集中随机采样70%数据做为训练数据构建多维偏好模型, 其余30%数据作为测试数据. 进而使用BNT中的推理工具进行推理, 以评分值为查询变量, 其余变量为证据变量, 以评分值后验概率最大时的取值作为最有可能的评分值. 将数据中用户评4或5分的电影记为用户所倾向的电影, 我们对推理所得的用户有倾向的电影和真实数据中用户有倾向的电影进行对比. 据此, 我们对模型的准确性, 覆盖率和F值进行了测试.

$Pre$ 表示准确性, 如公式(7)所示,  $num(inference)$ 是推理出用户倾向电影的数目,  $num(ture)$ 是推理出有倾向且实际中也有倾向的电影数目.

$$Pre = \frac{num(ture)}{num(inference)}. \quad (7)$$

$Cov$ 表示覆盖率, 如公式(8)所示,  $num(sample)$ 是实际中用户有倾向的电影的数目,  $num(ture)$ 是推理出有倾向且实际也有倾向的电影数目.

$$Cov = \frac{num(ture)}{num(sample)}. \quad (8)$$

$F$ 值反映了准确性和覆盖率的综合性能, 如公式(9)所示.

$$F = \frac{2 \times Pre \times Cov}{num(sample)}. \quad (9)$$

如图14、图15和图16所示, 随着数据量的增大, 准确性、覆盖率以及 $F$ 值均逐渐稳定, 这在一定程度上说明了本文所提方法的可扩展性. 同时, 随着数据量的上升, 应用本文模型对用户倾向电影预测的准确性逐渐稳定在0.57, 这表明使用本模型推理出的用户倾向符合实际的概率大于不符合实际的概率, 这在一定程度上说明了本文方法的有效性.

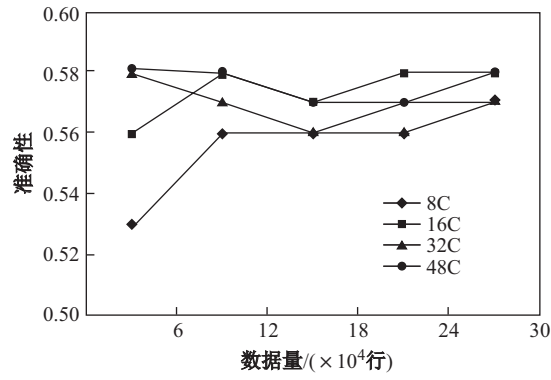


图 14 准确性

Fig. 14 Precision

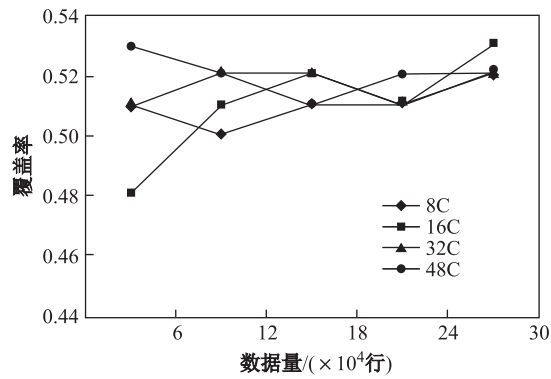


图 15 覆盖率

Fig. 15 Coverage

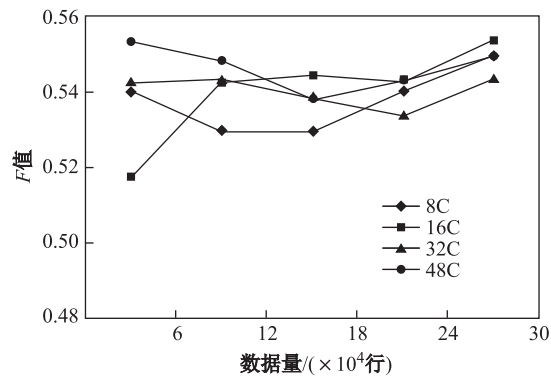


图 16 F 值

Fig. 16 F score

进一步对本文提出的模型 (MLVM) 与只含单隐变量的隐变量模型 (LVM)、SVD 和 LDA 的准确性、覆盖率和  $F$  值进行了比较, 分别如图 17、图 18 和图 19 所示. 随着数据量的增加, 使用本文模型对用户倾向电影进行预测的准确性高于其他三种模型; 覆盖率比 SVD 和 LVM 稍低, 但明显高于 LDA;  $F$  值高于其他三种模型. 这从一定程度上说明了本文模型的有效性.

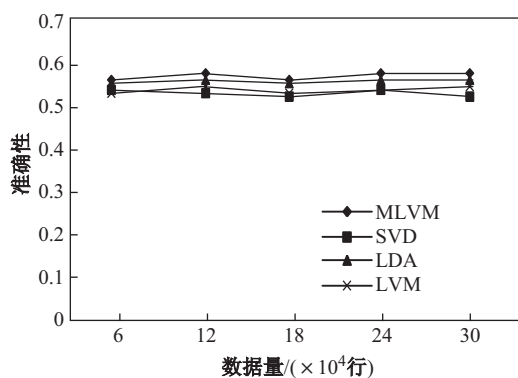


图 17 准确性对比

Fig. 17 Comparison of precision

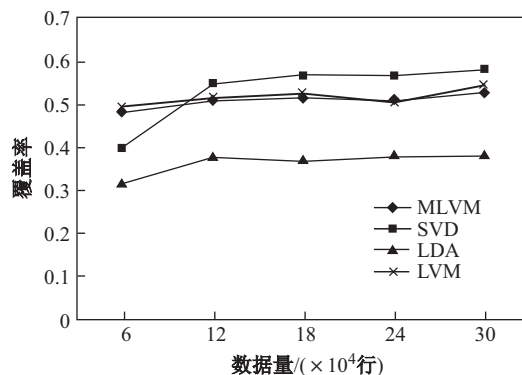
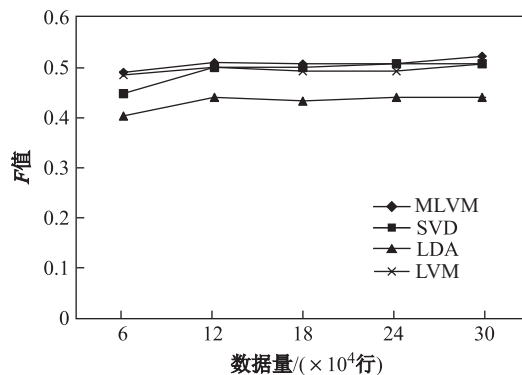


图 18 覆盖率对比

Fig. 18 Comparison of coverage

图 19  $F$  值对比Fig. 19 Comparison of  $F$  scores

## 5 总结与展望

本文从用户评分数据出发, 分析并描述了多维用户偏好, 以多个隐变量描述用户的多维偏好, 以含有多个隐变量的贝叶斯网来构建多维偏好模型, 并使用 Spark 计算框架实现模型构建方法, 建立在真实数据上的实验验证了本方法的效率和有效性. 本文的模型既可以表达多个维度的用户偏好, 又可以描述用户评分数据中各属性间任意的依赖关系, 为后续的个性化等服务提供了更加准确的保证.

本文方法构建的多维偏好模型,能够描述多个维度的用户偏好以及评分数据各属性间不确定的依赖关系。但是,本文方法只能在静态数据上进行建模,评分数据是动态变化的,以增量的方式构建偏好模型仍需进一步探索。同时,用户偏好也是动态变化的,建立动态模型描述,对变化的偏好进行估计,是我们将要开展的研究工作。

### [参 考 文 献]

- [1] GROUPLENS RESEARCH. MovieLens Dataset [EB/OL]. [2017-08-20]. <http://grouplens.org/datasets/movielens/>.
- [2] KASSAK O, KOMPAN M, BIELIKOVA M. User preference modeling by global and individual weights for personalized recommendation [J]. Acta Polytechnica Hungarica, 2015, 12(8): 27-41.
- [3] YIN H, CUI B, CHEN L, et al. Modeling location-based user rating profiles for personalized recommendation [J]. ACM Transactions on Knowledge Discovery from Data, 2015, 9(3): 1-41.
- [4] YUAN Q, CONG G, MA Z, et al. Who, where, when and what: Discover spatio-temporal topics for Twitter users [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013: 605-613.
- [5] ZHAO K, CONG G, YUAN Q, et al. SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews [C]// IEEE, International Conference on Data Engineering. IEEE, 2015: 675-686.
- [6] JIANG B, LIANG J, SHA Y, et al. Retweeting behavior prediction based on one-class collaborative filtering in social networks [C]// Proceedings of the 39th International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval. ACM, 2016: 977-980.
- [7] 张连文, 郭海鹏. 贝叶斯网引论 [M]. 北京: 科学出版社, 2006.
- [8] DAPHNEKOLLER, NIRFRIEDMAN, 科勒, 等. 概率图模型 [M]. 北京: 清华大学出版社, 2015.
- [9] KIM J S, JUN C H. Ranking evaluation of institutions based on a Bayesian network having a latent variable [J]. Knowledge-Based Systems, 2013, 50: 87-99.
- [10] SCHÜTZ W, SCHÄFER R. Bayesian networks for estimating the user's interests in the context of a configuration task [C]// Proceedings of the UM2001 Workshop on Machine Learning for User Modeling, 2001: 13-17.
- [11] FRIEDMAN N. The Bayesian structural EM algorithm [C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 2013: 129-138.
- [12] ELIDAN G, FRIEDMAN N. Learning hidden variable networks: The information bottleneck approach [J]. Journal of Machine Learning Research, 2005, 6(6): 81-127.
- [13] JIN C, ZHANG Y, BALAKRISHNAN S, et al. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences [J]. Advances in Neural Information Processing Systems, 2016, 4(1): 16-24.
- [14] ZHAO G, QIANX, XIE X. User-service rating prediction by exploring social users' rating behaviors [J]. IEEE Transactions on Multimedia, 2016, 18(3): 496-506.
- [15] 高全力, 高岭, 杨建锋, 等. 上下文感知推荐系统中基于用户认知行为的偏好获取方法 [J]. 计算机学报, 2015(9): 1767-1776.
- [16] 王红兵, 孙文龙, 王华兰. Web服务选择中偏好不确定问题的研究 [J]. 计算机学报, 2013, 36(2): 275-285.
- [17] 史艳翠, 孟祥武, 张玉洁, 等. 一种上下文移动用户偏好自适应学习方法 [J]. 软件学报, 2012, 23(10): 2533-2549.
- [18] GAO R, YUE K, WU H, et al. Modeling user preference from rating data based on the bayesian network with a latent variable [C]// Proceedings of 17th International Conference on Web-Age Information Management, 2016: 3-16.
- [19] HUANG Y, BIAN L. A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet [J]. Expert Systems with Applications, 2009, 36(1): 933-943.
- [20] CHAPELLE O, ZHANG Y. A dynamic Bayesian network click model for web search ranking [C]// Proceedings of the 18th International Conference on World Wide Web, ACM, 2009: 1-10.
- [21] HUETE J, DE CAMPOS L M, FERNANDEZ-LUNA J M, et al. Using structural content information for learning user profiles [C]// Proceedings of 30th Special Interest Group on Information Retrieval, 2007: 38-45.
- [22] AUFFENBERG F, STEIN S, ROGERS A. A personalised thermal comfort model using a Bayesian network [C]// Proceedings of the 2015 International Joint Conference on Artificial Intelligence, 2015: 130-139.
- [23] YUE K, FANG Q, WANG X, et al. A parallel and incremental approach for data-intensive learning of Bayesian networks [J]. IEEE Transactions on Cybernetics, 2015, 45(12): 2890-2909.
- [24] TAMADA Y, IMOTO S, MIYANO S. Parallel algorithm for learning optimal Bayesian network structure [J]. Journal of Machine Learning Research, 2011, 12(7): 2437-2459.
- [25] MIT. FullBNT [CP/OL]. [2017-08-20]. <http://www.cs.ubc.ca/~murphyk/Software/BNT/FullBNT-1.0.4.Zip>.

(责任编辑: 李万会)