

# 基于关联关系的仿真模型实时智能推荐方法

范国超<sup>1)</sup>, 许承东<sup>1)</sup>✉, 胡春生<sup>2)</sup>, 宋 丹<sup>1)</sup>

1) 北京理工大学宇航学院, 北京 100081 2) 宁夏大学机械工程学院, 银川 750021  
✉ 通信作者, E-mail: xucd@bit.edu.cn

**摘 要** 当全球导航卫星系统(global navigation satellite system, GNSS) 分布式仿真环境中共享的模型数量非常多时, 检索模型和配置仿真任务将成为一个比较复杂的工程. 为提高仿真模型选取和仿真任务配置的效率, 设计了一套针对 GNSS 分布式仿真环境中仿真模型的实时智能推荐方法. 方法中首先定义了模型关联关系和接口形状的概念, 然后提出了一种条件约束下的频繁模式树(FP-tree) 结构, 并从理论上分析了该结构在检索任务量方面的减少程度. 设计并推导了模型关联关系度的计算方法, 以及整套智能推荐方法的运行流程. 推荐方法在 GNSS 分布式仿真环境中进行了仿真验证. 仿真结果与传统智能推荐方法做对比分析, 分析结果表明, 该方法针对仿真模型推荐时运行时间短, 推荐结果准确度高, 能够实时为用户推荐合适的模型.

**关键词** 全球导航卫星系统; 分布式仿真; 智能推荐; 关联关系  
**分类号** TP391.9

## Real-time intelligent recommendation method of a simulation model based on incidence relation

FAN Guo-chao<sup>1)</sup>, XU Cheng-dong<sup>1)</sup>✉, HU Chun-sheng<sup>2)</sup>, SONG Dan<sup>1)</sup>

School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China  
School of Mechanical Engineering, Ningxia University, Yinchuan 750021, China  
✉Corresponding author, E-mail: xucd@bit.edu.cn

**ABSTRACT** With the availability of a large number of sharing models, model search and task design would be an extremely complex project in the global navigation satellite system (GNSS)-distributed simulation environment (GDSE). For improving the efficiency of model search and task design, a real-time intelligent recommendation method was designed for GDSE. Based on the characteristics of the simulation model, the incidence relation and interface shape of the model were defined in the method and a conditional frequent pattern tree (FP-tree) structure was designed to further improve the retrieval efficiency. The effect of the conditional FP-tree structure was proved theoretically. Then, the calculation method of the model incidence relation degree was proposed and derived based on the Bayesian statistical method. The entire processing of the intelligent recommendation method was designed for implementing it in GDSE. Hence, to check the effect of the real-time intelligent recommendation method, it was implemented in GDSE. Compared with the simulation result of the traditional recommendation method, the model intelligent recommendation method is proved to have a shorter running time and a high accuracy on simulation model recommendation. The computing capability and real-time performance are proved through the simulation. It is demonstrated that the intelligent recommendation method is efficient and flexible for GDSE.

**KEY WORDS** global navigation satellite system; distributed simulation; intelligent recommendation; incidence relation

全球导航卫星系统(global navigation satellite system, GNSS) 分布式仿真环境(GNSS distributed simula-

tion environment, GDSE) 是最近被提出的一种 GNSS 仿真环境. 该环境在面向互联网的前提下, 结合了广义云计算( cloud computing, CC) 核心理念——“对计算资源进行共享”<sup>[1-7]</sup>和面向服务架构( service oriented architecture, SOA) 技术<sup>[8-10]</sup>, 实现了多开发人员或用户发布并共用仿真模型或资源, 达到研究新算法、新设备和扩展新领域的目的. 李伯虎等<sup>[11-13]</sup>在研究中验证了结合云计算的仿真环境可以有效地组织数据资源、模型资源、计算资源和存储资源, 进而加速计算过程, 实现资源共享、资源管理、资源配置、并行计算、协同工作、信息服务和容错迁移. 该仿真平台具有高度的共享和重用能力, 能够减少重复开发和投资, 具备更高的仿真能力. 胡春生等<sup>[14-16]</sup>在结合云计算的分布式仿真环境研究中提出了一种服务化分布式仿真环境的基本构架和实施方法, 初步分析了模型管理的相关问题和研究思路, 但并没有完全达到智能化管理的目的. 随着服务化仿真平台的发展和应用, 仿真模型数目会以指数形式增大, 并且应用场景和仿真需求不同, 需要构造的仿真流程的差异也较大, 如何在大规模数量的模型中迅速找到合适仿真模型是一个十分重要的问题. 利用计算机推理自动检索并推送给用户恰当的模型可以避免用户多次手动检索模型.

目前针对计算机智能推荐方法主要基于数据挖掘技术设计的, 包括决策树、聚类、回归启发式方法、神经网络/关联规则挖掘等<sup>[17]</sup>, 这些方法在大规模数据挖掘和分析中具有很好的效果, 并且多数方法已经应用到了推荐系统上, 但这些方法针对的推荐系统主要应用领域为图书、文档、电影、商品等<sup>[17-20]</sup>, 被推荐对象为完全独立的个体, 互相基本没有相关关系. GNSS 分布式仿真环境中模型之间具有紧密的联系, 这种联系使得模型的推荐工作与传统推荐方法有着巨大的差别. 仿真任务中通常为 10 个以上模型的集合, 使用传统数据挖掘方法在仿真任务的大规模数据记录中提取信息两模型间关联关系时准确度会明显降低. 仿真模型智能推荐方法的数据检索和挖掘的过程与传统的模式也有很大的区别, 传统大数据挖掘过程通常是根据一定的支持度或置信度阈值检索出相应的集合得到其中元素的关联关系, 而模型的智能推荐工作中将先形成候选模型, 再依据候选模型检索得到统计结果并排序. 另外仿真平台中的模型还具有多种特性: 模型之间的连接和模型本身性质具有紧密的联系; 同一研究领域的用户设计仿真任务会受到学科内容约束, 在选择仿真模型时具有一定的相似性; 在不同应用场景和需求下, 仿真任务设计所需的模型具有一定的差异性; 智能推荐方法需要在用户配置任务时及时为用户提供参考模型. 上述性质对模型的智能推荐提出了一些新的需求, 如实时性、智能性、可配置性等.

本文从服务化模型基本性质出发, 提出了模型关联关系和接口形状的概念, 定义并推导了衡量模型关联关系的指标, 在关联规则的研究基础上提出了一种针对 GNSS 分布式仿真环境中仿真模型的智能推荐方法, 方法中首先在贝叶斯分析方法的基础上提出了模型关联关系度的计算方法, 然后设计了一种条件约束下 FP-tree 数据存储结构以及推荐方法工作流程, 并从理论上推理并讨论了该方法中部分工作的任务量等指标. 最后利用 200000 组虚拟仿真任务记录数据对方法进行仿真验证, 对模型智能推荐方法的各项性能进行测试, 并与基于关联规则的数据挖掘方法进行对比分析. 分析结果表明该方法在针对仿真模型推荐时运行时间短, 推荐结果准确度高, 具有很好的实施效果.

## 1 模型关联关系和接口形状

不同仿真环境中, 仿真模型和仿真任务流程的概念差别很大, 因此在定义模型关联关系和接口形状之前, 首先明确了这两个概念. 模型关联关系和接口形状是本文提出的两个概念, 仿真平台以模型关联关系和接口形状为基础对仿真模型智能化管理, 有助于仿真模型智能推荐方法的实现和推荐方法效率的提高.

### 1.1 仿真模型

大型仿真系统(包括数学仿真、半物理仿真等)是对系统级结构或平台的功能和性能上的模拟实现. 由于整体的数学或结构上的仿真很难直接实现, 通常将其分解为若干个既相对独立又相互联系的子系统(包含一个或多个功能单元), 这种子系统更方便使用数学模型或物理模型表示出来, 被用于表示的模型即为仿真模型. 不同系统的仿真可能包含相同功能的子系统, 即可使用相同的模型参与仿真.

在 GNSS 分布式仿真环境中, 利用上述思想, 用户将能够实现部分功能的仿真模型依据接口规则共享到平台上. 在有仿真需求时, 根据需求用户可以挑选共享的模型组合形成能够实现特定系统功能的仿真任务.

### 1.2 仿真任务流程

用户在利用 GNSS 分布式仿真环境构建仿真任务时, 需要调用仿真模型, 配置模型之间的输入与输出关系, 配置完成后形成了一组具有关联关系的模型集合, 这个包含配置信息和模型关联关系的模型集合即为仿真任务流程, 用户可以直接运行仿真任务流程进行相应的仿真任务, 获得最终想要的结果.

### 1.3 模型关联关系

在 GNSS 分布式仿真环境中, 数学方法、硬件设备等资源均可虚拟化为仿真模型, 并经过一定形式的封装后形成服务化模型. 这种服务化模型是完全独立的个体, 能够完成一定的特定功能, 并通过统一标准的接

口协议配置输入输出信息与其他仿真模型建立联系. 在多个仿真模型组合形成仿真任务时,仿真模型间需要建立一种特殊的关系来确定模型之间资源的传递(如数学仿真模型的输入输出参数信息等)这种关系即为仿真模型关联关系.

通过构建仿真任务,本文发现能够与同一仿真模型建立关系的几个模型构成的关系在一定程度上不是等价的,即他们与同一模型的亲疏程度有所不同,这种亲疏程度表现出了用户在建立仿真任务时的习惯偏好. 本文将这种模型之间的亲疏程度定义为模型关联关系度,将模型关联关系度进行量化分析,可以得到不同模型之间的亲疏程度排序.

确定模型关联关系度主要有两种因素,即用户操作信息和模型接口形状,故模型关联关系度可以定义为

$$d(M_A, M_B) = f(M_{user}, N_{interface}). \quad (1)$$

其中,  $M_A, M_B$  为两个建立关联关系的仿真模型,  $M_{user}$  为有关用户的信息统计函数,主要由用户已选模型、仿真任务记录、模型库等信息处理得到;  $N_{interface}$  为基于模型接口形状进行处理所得到的结果. 由于模型关联关系具有方向性,因此

$$d(M_A, M_B) \neq d(M_B, M_A). \quad (2)$$

关联关系度是一个对不同模型关联关系亲疏程度的量化指标. 因此与同一个模型建立关系的关联关系度为一个动态值,且存在归一化性质,即

$$\sum d(M_A, M_i) = 1. \quad (3)$$

其中,  $M_i$  为任意可以与  $M_A$  建立由  $M_A$  到  $M_i$  的连接关系的模型. 由上述性质可知,模型关联关系度是在一定条件下的概率问题的求解,且其概率密度符合后验分布,相应的求解方法可以贝叶斯统计为基础设计,即

$$d(M_A, M_i) = \frac{f(g(M_A, M_i) | M_i) \pi(M_i)}{\sum_{i=1}^m f(g(M_A, M_i) | M_i) \pi(M_i)}. \quad (4)$$

其中,  $g(M_A, M_i)$  为  $M_i$  的样本信息,  $f(g(M_A, M_i) | M_i)$  为  $M_i$  发生的概率密度函数,  $m$  为候选模型个数.

### 1.4 接口形状

仿真模型在封装后以统一的接口对外表现,主要包括输入接口、输出接口和初始化接口三类,其中输入接口和输出接口是用于在模型之间建立关系的,但不是任意模型之间都能够建立起模型关系,通常对于数学模型来讲,只有输入输出之间存在相同的量纲关系时才能够建立连接关系.

由于仿真模型可能包含多个输入或输出,所以本文规定,任意一个输入或输出均为一个模型接口,每个模型接口根据量纲和资源属性确定其接口形状. 在不同模型之间,只有含有相同形状的输出接口和输入接口两个模型才能建立关联关系,即:

$$\exists R(M_A, M_B) \Leftrightarrow \text{Out}(M_A) \cap \text{In}(M_B) \neq \emptyset. \quad (5)$$

式中,  $\text{Out}(A)$  表示  $M_A$  的输出接口形状的集合,  $\text{In}(B)$  表示  $M_B$  的输入接口形状的集合,若二者存在上述关系,则可以建立由  $M_A$  到  $M_B$  的连接关系.

接口形状池是指所有接口形状的集合,任意仿真模型的输入和输出的接口均为接口形状池的子集,如图 1. 将接口形状统计在一个集合下,便于对模型进行检索,给出具有相同接口形状的模型,建立模型间的关联关系. 因此,基于接口形状的检索可初步形成候选模型集合,记为接口形状池,这样将为挑选候选模型减少大量的检索工作,加快检索速度.

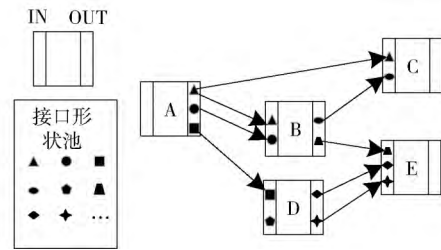


图 1 接口形状及接口形状池  
Fig. 1 Interface shape and its pool

## 2 智能推荐方法设计

仿真模型智能推荐方法结合用户使用模型的基本规律和模型的接口形状,根据当前用户所选择模型,智能推算下一个可能用到的模型,为用户设计仿真任务提供参考. 该方法主要依靠模型本身的基本属性和统计学方法,快速求取模型关联关系度并以此排序. 同时在数据挖掘过程中需要将相关方法依据仿真模型特性重新设计.

图 2 是仿真任务智能推荐模块在 GNSS 分布式仿真环境中的基本架构,该模块根据语义管理库中的接口形状池、模型库、仿真任务记录池以及人机交互界面中用户当前选择的部分模型进行检索和推理,最终给出一组经过排序的用户可能即将需要用到的模型集合. 图中可以看出,基于接口形状池的初步检索形成了候选模型集合,在依据模型关联关系度的计算和排序,对候选模型进行二次筛选和排序,最终确定了被推荐模型的有序集合.

### 2.1 模型关联关系度计算方法

模型关联关系度主要根据以往配置信息记录,结合权重的配置进行计算和比较分析. 由公式(1)可知,关联关系度需结合用户信息和接口形状两个方面进行求解.

(1) 用户信息部分求解.

用户信息部分主要包括对模型库、用户当前操作内容、仿真任务记录等相关信息的统计计算. 在统计过程中采用支持度的计算方法获得对应结果.

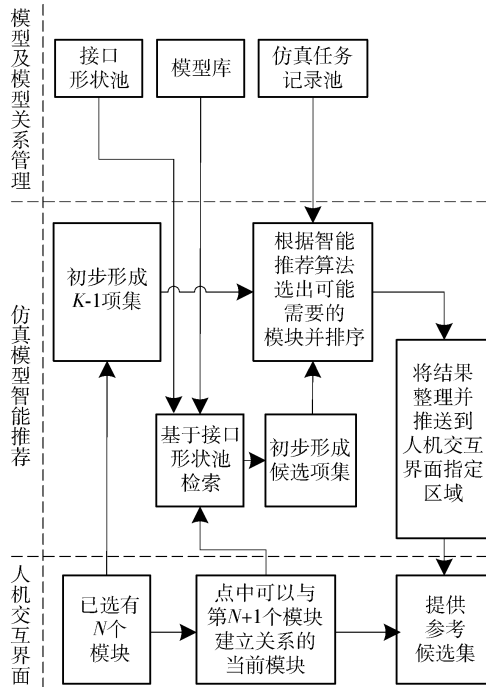


图2 仿真模型推荐算法在平台中的基本交互

Fig.2 Basic interaction of the simulation model illustration algorithm in the platform

故对用户信息部分的统计计算为

$$M_{user} = \sup(F_n \cup M_k). \quad (6)$$

其中,  $\sup()$  为对应模型集合在仿真任务记录中的支持度计数,  $F_n$  表示用户已经配置完的  $n$  个模型.

公式(6) 在模型检索过程中会出现项集检索计数结果为 0 的可能性, 这时需要项集中排除部分权重较低的模型重新检索, 同时检索得到的计数结果需要给一个权重系数. 故公式(6) 可修改为

$$M_{user} = \sup((F_n - N_j) \cup M_k) \times Q_1(N_j). \quad (7)$$

其中,  $Q_1(N_j)$  为事务项集中缺失一些模型后的剩余权重因子,  $N_j$  是在用户已经配置完成的  $n$  个模型中需要去掉的部分模型(可能是一个或多个).

缺失模型后剩余权重因子需要通过分别求取包含用户已选的全部模型和去掉一部分模型的两个集合的支持度, 进而通过相应的计算公式得到剩余模型的权重值. 因此  $Q_1(N_j)$  的计算方法为

$$Q_1(N_j) = 1 - \frac{\sup(F_n)}{\sup(F_n - N_j)}. \quad (8)$$

$$d(M_A, M_{(n)}) = \frac{a_1^{(n)} \sup((F_n - N_j^{(n)}) \cup M_{(n)}) \left[ 1 - \frac{\sup(F_n)}{\sup(F_n - N_j^{(n)})} \right] + a_2^{(n)} \frac{N_c^{(n)}}{N_{A(out)}} \frac{N_c^{(n)}}{N_{(in)}}}{\sum_{i=1}^m \left\{ \left( a_1^{(i)} \sup((F_n - N_j^{(i)}) \cup M_{(i)}) \left[ 1 - \frac{\sup(F_n)}{\sup(F_n - N_j^{(i)})} \right] + a_2^{(i)} \frac{N_c^{(i)}}{N_{A(out)}} \frac{N_c^{(i)}}{N_{(in)}} \right) \right\}}. \quad (13)$$

上述公式对新加入的模型可能会有一定的影响, 导致新加入模型始终不能被检测到, 因此为了对新加入模型有一定的排序保护, 将公式(13) 修改为

由式(7) 和式(8) 可得

$$M_{user} = \sup((F_n - N_j) \cup M_k) \cdot \left( 1 - \frac{\sup(F_n)}{\sup(F_n - N_j)} \right). \quad (9)$$

(2) 接口形状部分求解.

接口形状权重因子在计算过程中主要依据前后模型本身的接口数目和建立关系的接口数目来确定, 其计算公式如下,

$$N_{interface} = \frac{N_c}{N_{1(out)}} \cdot \frac{N_c}{N_{2(in)}}. \quad (10)$$

其中,  $N_c$  表示建立连接的接口形状数目,  $N_{1(out)}$  表示建立连接模型中的前一个模型输出接口的数目,  $N_{2(in)}$  表示建立连接模型中后一个模型的输入接口的数目.

(3) 关联关系度计算.

根据关联关系度的定义和相关性质以及贝叶斯统计方法, 可定义第  $n$  个候选模型对应的关联关系度计算方法为

$$d(M_A, M_{(n)}) = \frac{(a_1^{(n)} M_{user}^{(n)} + a_2^{(n)} N_{interface}^{(n)}) / F_{Total}}{\sum_{i=1}^m ((a_1^{(i)} M_{user}^{(i)} + a_2^{(i)} N_{interface}^{(i)}) / F_{Total})}. \quad (11)$$

其中,  $n$  表示候选集合中的第  $n$  候选模型,  $m$  表示候选集合中共有  $m$  个模型,  $F_{Total}$  为所有模型与当前模型发生关联的总次数.  $a_1$  为接口形状权重系数,  $a_2$  为缺失模型后剩余权重系数, 且  $a_1 + a_2 = 1$ .

$a_1$  和  $a_2$  的值主要由用户进行配置, 希望模型的全部接口能够快速建立连接的用户可以将  $a_1$  的值配置的相对更高一些, 而希望参考已有设计任务记录的用户可以将  $a_2$  的值配置的相对更高一些. 若  $a_1 = 0$ , 则表示用户完全依据任务设计流程记录对候选模型排序; 若  $a_2 = 0$ , 则表示完全不考虑缺失模型对排序结果的影响, 计算结果将主要受两个连接模型的接口形状数目和可连接接口形状数目影响.

公式(11) 可化简为

$$d(M_A, M_{(n)}) = \frac{a_1^{(n)} M_{user}^{(n)} + a_2^{(n)} N_{interface}^{(n)}}{\sum_{i=1}^m (a_1^{(i)} M_{user}^{(i)} + a_2^{(i)} N_{interface}^{(i)})}. \quad (12)$$

由公式(9)、(10) 和(12) 可得

$$d(M_A, M_{(n)}) = \frac{b + a_1^{(n)} \sup((F_n - N_j^{(n)}) \cup M_{(n)}) \left[ 1 - \frac{\sup(F_n)}{\sup(F_n - N_j^{(n)})} \right] + a_2^{(n)} \frac{N_c^{(n)}}{N_{A(out)}} \frac{N_c^{(n)}}{N_{(in)}}}{b + \sum_{i=1}^m \left\{ a_1^{(i)} \sup((F_n - N_j^{(i)}) \cup M_{(i)}) \left[ 1 - \frac{\sup(F_n)}{\sup(F_n - N_j^{(i)})} \right] + a_2^{(i)} \frac{N_c^{(i)}}{N_{A(out)}} \frac{N_c^{(i)}}{N_{(in)}} \right\}}. \quad (14)$$

$b$  为模型在刚加入平台中时设置的初始数目,为了不影响后续模型的统计信息,  $b$  的取值不宜过大,通常选取在 100 左右,用户可根据个人习惯进行配置。

### 2.2 条件 FP-tree 数据存储结构设计

在对大量仿真任务记录进行检索时,采用频繁模式树(FP-tree)数据记录结构可以加快检索效率。FP-tree是做频繁项集等数据挖掘工作常用的数据记录模式<sup>[21]</sup>,通常存储的不同事务之间会有若干个相同的项,因此它们在 FP-tree 上可能会有部分路径重叠。路径重叠的越多,使用 FP-tree 结构获得的压缩的效果越好<sup>[22]</sup>,提高对存储事务检索的速度就越明显,通过支持度阈值或置信度阈值检索效率也就越高。但以仿真平台中的模型为对象做关联关系检索时,已知参数不再是支持度阈值或置信度阈值等约束量,而是各个候选模型分别对应的候选项集。这样在考察各个候选模型对应项集的支持度时均需单独检索 FP-tree,设每次 FP-tree 的检索任务量为  $L$ ,候选模型有  $m$  个,用户已选模型  $p$  个,则总检索任务量为

$$R_i = d_1 L + d_2 L + \dots + d_m L = \sum_{i=1}^m d_i L, \quad d_i \in [1, p-1]. \quad (15)$$

所以有

$$R_i \in [mL, m(p-1)L]. \quad (16)$$

在检索某一候选模型对应的项集时,其他候选模型对应的 FP-tree 结点和分支即成为多余检索量。本文在 FP-tree 的基础上,提出了固连结点的概念,在固连结点形成一种具有一定条件约束的 FP-tree 数据记录结构。固连结点是把与 FP-tree 根节点直接相连的节点换成固定的连接体,连接体由候选模型对应的节点和当前用户选中模型对应的节点构成。在固连结点之后的节点采用与传统 FP-tree 相同的构建方法,每个条件 FP-tree 的固连结点的数目由候选模型数目决定。这样在针对每个候选模型做 FP-tree 检索时,可以直接检索其对应固连结点的分枝,其余固连结点的分枝直接做剪枝处理。这样可以节省大量检索时间,提高推荐方法效率。

以固连结点为约束条件的 FP-tree 基本结构如图 3。图中 E、F、G 和 H 为根据接口形状池检索出的候选模型, D 为当前用户选中的模型, A、B 和 C 为用户已选的其他模型。

依据上述方法,假设第  $i$  个候选模型对应分支的检索任务量为  $l_i$ ,则有

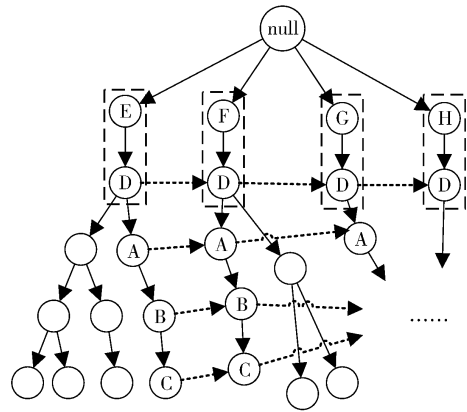


图 3 条件 FP-tree 数据结构

Fig. 3 Data structure of the condition FP-tree

$$L = \sum_{i=1}^m l_i. \quad (17)$$

每个候选模型对应的检索任务量为

$$r_i = d_1 l_1 + d_2 l_2 + \dots + d_m l_m, \quad d_i \in [1, p-1]. \quad (18)$$

根据  $d_i$  取值范围可得

$$r_i \in [l_i, (p-1)l_i], \quad i \in [1, m]. \quad (19)$$

在检索过程中,  $m$  个候选模型检索任务总量为

$$R_i \in \left[ \sum_{i=1}^m l_i, (p-1) \sum_{i=1}^m l_i \right] = [L, (p-1)L], \quad i \in [1, m]. \quad (20)$$

若  $d_1 = d_2 = \dots = d_m$ , 则有

$$R_i = \frac{R_i}{m}. \quad (21)$$

对比公式(16)可见加入固连结点约束后的 FP-tree 数据存储结构在候选模型项集检索方面大幅降低了检索任务量,由公式(20~21)可以看出,候选模型越多检索任务量降低的效果将越明显。

### 2.3 推荐方法流程

智能推荐方法的工作主要是对模型库、接口形状池、仿真任务记录池等数据记录的一系列检索和计算,将得到的结果依照模型关联关系度排序后提供给用户,其主要工作流程如图 4。

① 根据当前选中的模型,以及接口形状池的记录,在模型库中提取出所有能够与当前选中模型建立连接关系的模型,组成候选项集  $P_{(1 \times m)}$ 。

② 根据候选模型和当前模型的接口数目进行计算,得到所有候选模型对应的接口形状权重  $N_{interface}$ 。

③ 将候选项集  $P_{(1 \times m)}$  和当前需要建立连接的模型  $q_n$  构建固连结点矩阵  $[P_{(1 \times m)}^T, [o_n \dots o_n]_{(1 \times m)}^T]$ ,

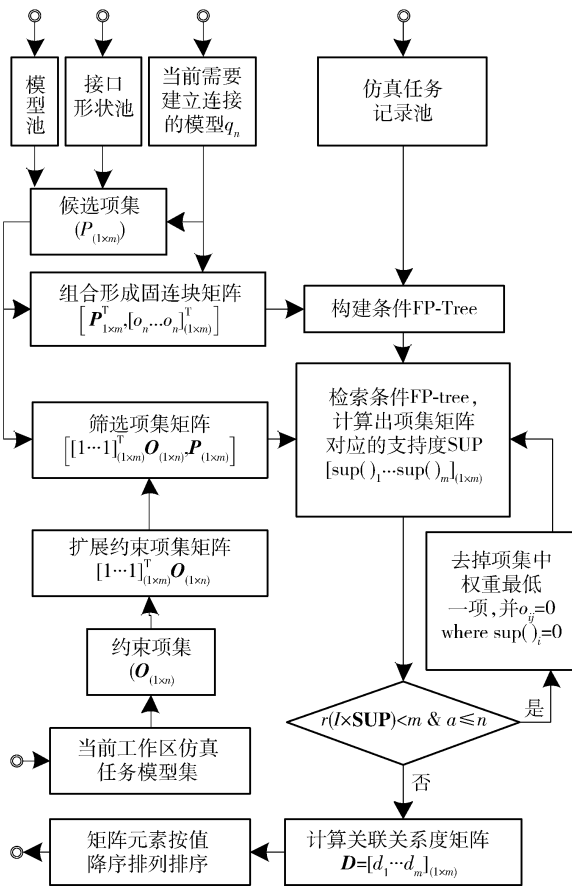


图4 智能推荐算法工作流程

Fig.4 Intelligent recommendation algorithm workflow

并在固连结点的约束下检索仿真任务池,构建条件FP-tree.

④ 根据用户已选模型形成的约束项集矩阵  $O_{(1 \times n)}$  和候选项集矩阵  $P_{(1 \times m)}$  构建成仿真模型筛选项集矩阵  $[[1 \dots 1]_{(1 \times m)}^T, O_{(1 \times n)}, P_{(1 \times m)}]$ .

⑤ 检索条件FP-tree 根据支持度计算方法得到支持度矩阵  $SUP = [sup()_1 \dots sup()_m]_{(1 \times m)}$ , 并判定结果  $I \times SUP$  是否满质, 若不满质, 去掉  $sup()_i = 0$  处的权重最小的模型, 并设置  $o_{ij} = 0$ , 重新计算.

⑥ 获得满质的  $I \times SUP$  后, 根据模型关联关系度的计算方法, 得到候选模型项集对应的关系度矩阵  $D = [d_1 \dots d_m]_{(1 \times m)}$ , 并将矩阵元素降序排列, 依照排列顺序, 推送模型到人机交互界面.

### 3 仿真结果分析

为检验上述方法的可用性, 在局域网环境中, 构建了一个GNSS分布式仿真环境原型系统, 仿真平台环境搭建如图5, 在该环境中, 模型库、仿真任务记录池和接口形状池分别存储于对应的数据存储服务器中, 仿真终端、推荐算法服务器等服务终端以面向服务框架(SOA)架构面向互联网构建, 其中推荐算法模块单

独配置到一台服务器上, 网络带宽为100 Mbps.

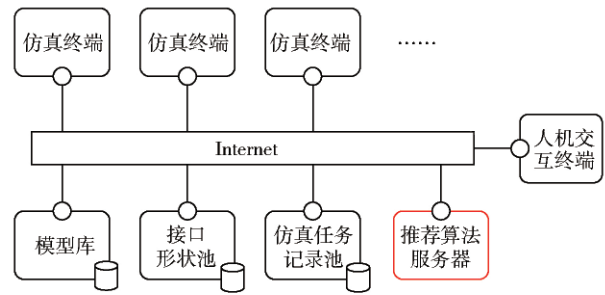


图5 仿真环境基本架构

Fig.5 Architecture of the simulation environment

测试前首先人工生成了250个仿真模型(包括星历文件读取、空间环境仿真、观测数据仿真、用户位置仿真、用户定位解算等)和200000条仿真任务记录. 在人机交互终端进行任务设计, 利用本文智能推荐方法在仿真任务设计过程中对下一步模型进行预测, 分析其方法运行速度等指标, 并采用传统数据挖掘方法(以关联规则方法为例)对同样任务处理用于对比分析, 另外为测试条件FP-tree对模型智能推荐方法运行性能的提升, 单独记录了不采用条件FP-tree数据存储结构的智能推荐方法的工作时间. 为保证统计结果可靠性, 使用蒙特卡洛方法分别针对200种不同仿真任务测试, 得到用户在已配置4个模型时各方法的平均工作时间如图6.

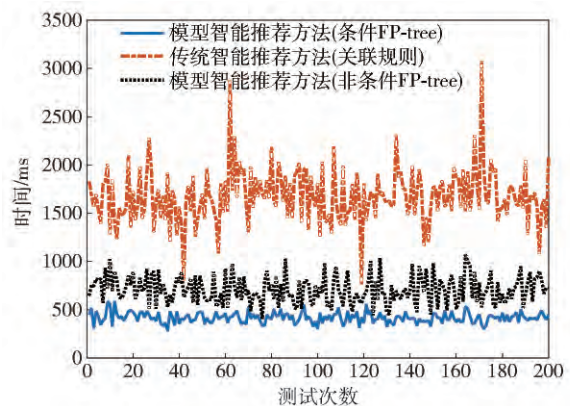


图6 推荐算法运行时间

Fig.6 Running time of the recording method

图6中可以看出, 使用本文中的模型智能推荐方法工作时间维持在500ms左右, 而使用传统的基于关联规则的数据挖掘方法在上述任务中工作时间为1500~2000ms. 可见模型智能推荐方法具有更高的效率, 基本能够达到实时反馈. 另外不使用条件FP-tree时, 模型智能推荐方法的工作时间达到800~1000ms, 时间增长了近一倍, 可以看出条件FP-tree在实际工作中可以明显提高推荐方法的工作效率.

目前通常进行的仿真任务包含模型一般为8~15

个,为测试智能推荐方法与用户已经选择模型的数目之间的关系,分别以不同用户已配置模型数目(1~15个)进行蒙特卡洛测试,得到用户已选模型的数量与推荐方法平均工作时间的关系如图7所示。

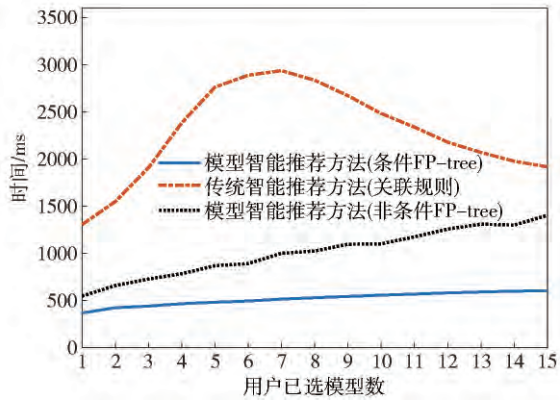


图7 推荐方法运行时间与用户已配置模型数目的关系

Fig.7 Relationship between the running time and configured model number

图7中可以看出传统智能推荐方法在开始时随已选模型数目的增大急剧增加,在8个左右之后会有所减少,这主要是因为传统方法中不同模型数目组成的项集数目只是简单的组合关系,项集数目具有先增多后减少的特点,对应的运行时间也会有同样的影响。而模型智能推荐方法的候选模型是通过接口形状筛选的,不受组合数目的影响,因此其运行时间虽然会随着用户已配置的模型数目的增加有所增大,但始终维持在500ms左右。另外,对比图中未采用条件FP-tree的运行时间的增加情况来看,智能推荐方法增加的工作时间也主要为FP-tree检索时间。

根据用户在人机交互端已配置4个模型的推荐结果,统计各个方法得到的推荐模型的准确度。统计过程中,本文规定:推荐集中包含用户所需模型即为有效,若不包含即为无效。推荐模型集中模型个数由用户确定:假设用户设置为x个,则取关联关系较大的前x个模型为推荐模型。得到各方法对应的推荐模型准确度如图8所示。

图8中可以看出,当仅取1个推荐模型时,模型智能推荐方法的准确度为40%~50%,推荐模型个数为10个时,准确度已达到95%~98%,接近完全准确。而采用传统推荐方法所得结果的准确度始终在40%以下,甚至当推荐模型为2个或以下时准确度已低于20%。另外,无论是否采用条件FP-tree存储结构,其检索结果的准确度不受影响,可见数据存储结构并不影响检索和计算的结果。

分别以不同仿真任务记录数目进行分析,得到推荐算法的平均工作时间如图9所示。

图中模型智能推荐方法的工作时间会随仿真任务

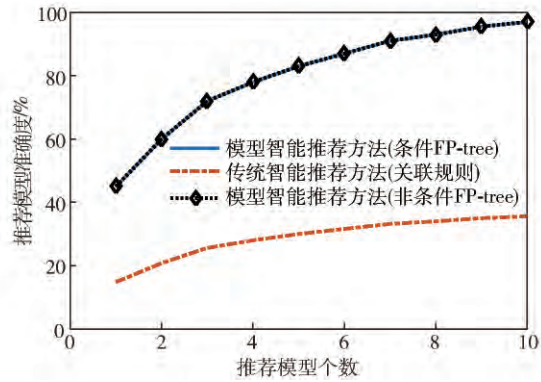


图8 推荐方法准确度

Fig.8 Accuracy of the recording method

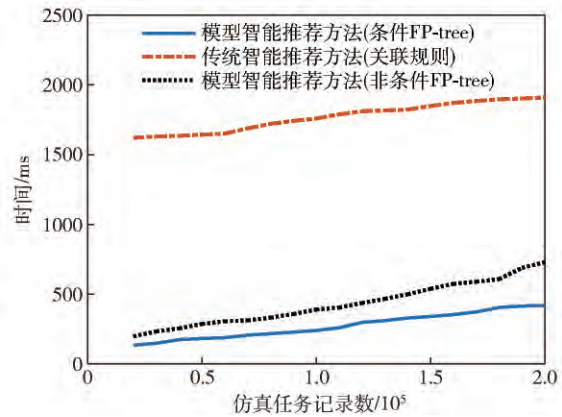


图9 推荐方法运行时间与仿真任务记录数目的关系

Fig.9 Relationship between the running time and simulation task records

记录数目的增加会有所增大,但总体运行时间一直保持在500ms以内。而使用传统数据挖掘方法所消耗的时间均在1500ms以上,随着仿真任务记录的增加,方法所需时间也会增大。未使用条件FP-tree时间增加的趋势更加明显,可以推测在增长到一定程度后,该方法所消耗的时间可能会过长,甚至不满足方法的实时性需求。

上述结果表明,本文设计的仿真模型智能推荐方法与传统方法比较,运行时间更少,并且不会随着用户已选模型、仿真任务记录等参量的变化有较大的变动。另外经过测试,条件FP-tree数据存储结构对减少模型检索时间具有很好的作用。

### 4 结论

根据GNSS仿真的基本原理,在其服务化共享仿真研究的基础上,结合仿真模型的基本特性,针对GNSS分布式仿真环境中模型关联关系进行推理分析,提出了一种针对平台中仿真模型的智能推荐方法,该方法结合数据挖掘的相关技术,利用仿真模型的基本特性,基本能够实现根据现有模型及仿真任务记录自

主推理 给出下一步用户所需仿真模型的参考模型集合,同时计算对应的模型关联关系度并排序,基本实现了仿真模型的实时智能推荐功能。仿真验证结果和对比分析表明,该方法针对仿真模型推荐时运行时间短,推荐结果准确度高,方法具有很好的效果,且可行性较强。符合 GNSS 仿真任务设计中模型推荐的实时性的功能和性能需求。在后续研究中将针对方法优化,提高计算效率,减少计算空间,并结合模型的语义管理进行优化设计,进一步提高智能推荐算法在 GNSS 分布式仿真环境中的性能。

#### 参 考 文 献

- [1] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. *Commun ACM*, 2010, 53(4): 50
- [2] Mell P, Grance T. *The NIST Definition of Cloud Computing*. NIST Special Publication 800-145, 2011
- [3] Baun C, Kunze M, Nimis J, et al. *Cloud Computing Web-based Dynamic it Services*. Berlin: Springer-Verlag, 2011
- [4] Miller M. *Cloud Computing: Web-Based Applications that Change the Way You Work and Collaborate Online*. Indianapolis: QUE Publishing Company, 2008
- [5] Chen T. On cloud computing theory and its technique. *J Chongqing Jiaotong Univ Soc Sci*, 2009, 9(4): 104  
(陈涛. 云计算理论及技术研究. 重庆交通大学学报(社会科学版), 2009, 9(4): 104)
- [6] Buyya R, Ranjan R, Calheiros R N. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities // *International Conference on High Performance Computing & Simulation*. IEEE, Leipzig, 2009: 1
- [7] Fang B L, Yin X, Tan Y, et al. The contributions of cloud technologies to smart grid. *Renewable Sustainable Energy Rev*, 2016, 59: 1326
- [8] Vrba P, Marik V, Siano P, et al. A review of agent and service-oriented concepts applied to intelligent energy systems. *IEEE Trans Ind Inf*, 2014, 10(3): 1890
- [9] Tsai W T, Bai X Y, Huang Y. Software-as-a-service (SaaS): perspectives and challenges. *Sci China Inf Sci*, 2014, 57(5): 1
- [10] Wu D Z, Greer M J, Rosen D W, et al. Cloud manufacturing: strategic vision and state-of-the-art. *J Manuf Syst*, 2013, 32(4): 564
- [11] Li B H, Chai X D, Hou B C, et al. Networked modeling & simulation platform based on concept of cloud computing: cloud simulation platform. *J Syst Simul*, 2009, 21(17): 5292  
(李伯虎, 柴旭东, 侯宝存. 一种基于云计算理念的网络化建模与仿真平台“云仿真平台”. 系统仿真学报, 2009, 21(17): 5292)
- [12] Huang A X, Feng X W, Li J S, et al. Aviation simulation architecture based on cloud computing platform. *J Syst Simul*, 2011, 23(Suppl1): 106  
(黄安祥, 冯晓文, 李劲松. 基于云计算平台的航空兵训练仿真体系结构. 系统仿真学报, 2011, 23(增刊1): 106)
- [13] Armbrust M, Fox A, Griffith R, et al. *Above the Clouds: A Berkeley View of Cloud Computing*. University of California Berkeley, 2009
- [14] Hu C S. *Study on Key Techniques of Integrated Design Environment for Complex Product Based on Service-oriented Architecture* [Dissertation]. Beijing: Beijing Institute of Technology, 2014  
(胡春生. 面向服务的复杂产品集成设计环境关键技术研究[学位论文]. 北京: 北京理工大学, 2014)
- [15] Hu C S, Xu C D, Fan G C, et al. A simulation model design method for cloud-based simulation environment. *Adv Mech Eng*, 2013(8): 1
- [16] Hu C S, Xu C D, Zhang P F, et al. Study on simulation model virtualization in cloud computing environment. *J Huazhong Univ Sci Technol Nat Sci*, 2012, 40(Suppl 1): 135  
(胡春生, 许承东, 张鹏飞. 云计算环境下仿真模型资源虚拟化研究. 华中科技大学学报(自然科学版), 2012, 40(增刊1): 135)
- [17] Park D H, Kim H K, Choi I Y, et al. A literature review and classification of recommender systems research. *Expert Syst Appl*, 2012, 39(11): 10059
- [18] Borràs J, Moreno A, Valls A. Intelligent tourism recommender systems: a survey. *Expert Syst Appl*, 2014, 41(16): 7370
- [19] Chen S N, Qian H Y, Gu J Y. A recommender system for mobile commerce based on relational learning // *9th International Workshop Multi-disciplinary Trends in Artificial Intelligence, MIWAI 2015*. Fuzhou, 2015, 9426: 415
- [20] Han J S, Kim G J. A method of intelligent recommendation using task ontology. *Cluster Comput*, 2014, 17(3): 827
- [21] Dai C Y, Chen L. An algorithm for mining frequent closed itemsets in data stream. *Phys Procedia*, 2012, 24: 1722
- [22] Tan P N, Steinbach M, Kumar V. *Introduction to Data Mining*. Beijing: China Machine Press, 2010: 363