

个性化学习应用中数据结构的设计

张昊 1927405160

1 引言

1.1 场景描述

在一些个性化学习应用中拥有大量的课程，这些课程由许多知识点组成。某一个课程包含若干知识点，知识点之间有一定的前驱、后继、父、子和兄弟关系。同时为了满足个性化学习的需求，若干知识点会组织成一个模块，若干模块组合成一类课程，提供给用户学习。为了支撑相关的应用，设计相关的数据结构。

1.2 背景分析

互联网的迅速发展不断改变着人们的学习和生活方式，同时也极大地拓宽了获取知识的渠道。然而，知识的碎片化现象也同时出现。单纯地从这些网络平台上获取知识来看，我们每个人获取的知识是相对离散的；网络上的各种慕课平台获取到的知识大多为单个或多个零碎知识点的视频及其组合资源的简单堆积，课程的一些知识点之间缺少较强的组织与关联。人们主动到各种慕课等相关的应用平台上根据自己的需求学习知识，这类应用也需要根据每个人的需求来个性化推荐课程，这样一来，知识点间的多样的关系如何在计算机中表示显得尤为重要。

在个性化推荐等 AI 系统中，知识的表示与管理是核心问题。知识表示包括一个系统，该系统提供到知识体的通路和对知识访问的手段，知识体即为存放在存储器中的数据结构^[1]。本文通过讨论其中数据结构的设计，从而支撑相关的应用。

为了解决这一问题，本文从这些个性化学习应用中知识点的关系和模型入手，参考知识图谱的理论，探讨表征知识点、模块、课程等在现实中关系的数据结构——知识点多关系图，并分析其逻辑结构以及存储结构。

2 设计思路与概念说明

2.1 概要设计

在个性化学习应用中的学习内容和资料的结构与特点主要可分为 3 个层次，其中课程是第一个层级，课程中又包含若干个章节知识点组成的模块，知识点是学习与教学中的基本单元，在相关的应用中不能再进一步细分。因而个性化学习应用中数据结构应该从知识点及其之间的关系的表示来展开。

知识点存在一定的结构关系,文献[2]提出了知识点之间具有的结构关系包括层次关系、依赖关系、蕴涵关系、参考关系和游离关系。本文主要以知识点的层次关系和依赖关系为主来设计数据结构。

从上面的分析可见,知识点是整个应用体系中最为基础且相对独立的部分,而知识点间存在着多样的关联关系。所以,我们很容易地将这种逻辑关系抽象成一种图的数据结构来表征。同时参考知识图谱的有关概念,从知识点的层次来描述和呈现知识点之间的关系。因此,基于知识图谱的理念,设计一种可以表示知识点的数据结构。采用知识图谱来呈现学科知识,能够对知识点及知识点间的关系进行结构化描述,能够更好地帮助学习者了解学科知识体系^[3]。

2.2 概念阐述

2.2.1 语义网络

语义网络是通过包含结点和连接结点之间的弧的有向图来描述对象、事件、概念以及动作的方法,由于它很自然,所以在人工智能中获得广泛应用^[1]。

在语义网络中,程序可以从感兴趣的任何结点出发,沿着某些弧到达相关联的结点,还可继续沿弧到达更远的结点,这种方法非常自然,类似于人类的联想记忆。但是,鉴于每个结点连接多条弧,当我们从开始结点出发后,如果没有很好的组织和强有力的搜索规则指引,就会很容易陷入无穷支路而无解。

2.2.2 知识图谱与多关系图

知识图谱(Knowledge Graph)是结构化的语义知识库,用于以符号形式描述世界中的概念及其相互关系,其基本组成单位是“实体—关系—实体”三元组,以及实体及其相关属性—值对,实体间通过关系相互联结,构成网状的知识结构^[4]。

知识图谱本身是一个具有属性的实体通过关系链接而成的网状知识库。从图的角度在本质上是一种概念上的网络,其中的顶点表示实体或概念,而实体间的各种关系则构成网络中的弧。值得注意的是,课程中的知识点为概念中的“实体”,这些知识点通过他们中间的关系来相互联结;而知识库中的“知识”(或者称作“事实”)是指这些知识点之间的关系。

从图论的角度,知识图谱也可以认为是一种多关系图,它由作为顶点的实体和作为不同类型边的关系组成^[5]。在实际应用中,经常使用**多关系图**(Multi-relational Graph)来简化知识图谱^[6]。在本文中,为了消除“知识”与“知识点”可能造成的概念上的混淆,使用简化的知识图谱——多关系图来描述这一数据结构,即顶点表示知识点,弧表示知识点之间的关系。

2.2.3 AOV 网与拓扑序列

在一个表示工程的有向图中，用顶点表示活动，用弧表示活动之间的优先关系，这样的有向图为顶点表示活动的网，称之为 AOV 网 (Activity on Vertex Network)。AOV 网中的弧表示活动之间存在的某种制约关系。

在一个具有 n 个顶点的有向图 $G = (V, E)$ 中， V 中的顶点序列 v_1, v_2, \dots, v_n 称为一个**拓扑序列**，当且仅当满足下列条件：若从顶点 v_i 到 v_j 有一条路径，则在顶点的拓扑序列中顶点 v_i 必在顶点 v_j 之前。

在这里，我们将知识点看作是有向图的顶点，知识点之间前驱后继的这种依赖关系作为弧，可以得到一个表示知识点的前驱后继关系的 AOV 网。显然，AOV 网是上面提到的表示知识点及其关系的多关系图的一个子图，且如果编排合理不会存在回路。这个 AOV 网的拓扑序列即为知识点的先修后续顺序。

2.2.4 知识树

树结构比较适合描述具有层次关系的数据，如祖先—后代、上级—下级、整体—部分以及其他类似的关系。很多实际问题抽象的数据模型是树结构。

知识树本质上就是一个层级式知识图，它表达了为实现某一组织目标的所有相关组织知识间的从属关系。

3 逻辑结构设计

本文将表征知识点及其之间的关系的的数据结构称为**知识点多关系图**，其逻辑结构将从如下几个方面进行阐述。

3.1 知识点及其属性

由于知识点在知识点的多关系图中处于基础的地位，在表示逻辑关系的层面上不可再分，是讨论这一数据结构的最小数据单位，故将知识点作为数据元素。

构成数据元素的最小单位称为数据项。对于知识点来说，仅用知识点名称无法全面描述知识点的特征；此外，根据知识图谱的有关概念，其中包含了知识点作为一种实体的相关属性一值对。所以需要设计其数据属性，来描述知识点的自身属性。由于本文是建立在已有知识点集合的基础上进行的数据结构的设计方法的探讨，故这里不展开讨论知识点内部的属性应该如何设计，仅将知识点抽象为包含有唯一标识符、名称和其他属性的结点（如图 1），其他属性可以根据实际应用场景来进行扩充。

唯一标识符
名称
其他属性

图 1 知识点及其属性表示图

3.2 知识点间的关系

在知识点的多关系图中，顶点（知识点）之间的关系不再是简单的关联关系，而根据不同的实际应用场景有了不同的含义。

知识点是学习与教学中传递与接收信息的基本单元。知识点总是处在一定的结构之中，各个知识点之间不是一个个的孤岛，而存在着结构上的关系。这些关系的种类不尽相同，共同组成了知识点之间的逻辑关系。在知识点多关系网中，这些关系映射为链接各个顶点的弧，与知识点共同构成网络。具体地，按照知识点间关系在现实中的逻辑意义，可以将知识点之间的关系主要划分为以下几类：层次关系、依赖关系和相关关系。相应地，网络中的各弧也被分成了不同的类别。

3.2.1 层次关系

层次关系指某些知识点可以由若干知识点聚合而成，具有一定从属关系知识点之间可以建立起具有树状结构的层次结构。下面先介绍在知识树中具体的层次关系的种类。

借用树中的有关概念，知识树中某知识点从属于另一知识点，这一关系称为**子关系**（Child）；某知识点可由其他知识点聚合而成，这一关系称为**父关系**（Parent）（如图 2）；同时直接从属于同一知识点的各知识点之间的关系记为**兄弟关系**。由于兄弟关系的确认可以通过判断它们的父级知识点是否相同直接得到，故在知识点多关系网中，层次关系更关注父关系和子关系。

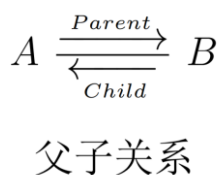


图 2 父子关系示意图

将知识点多关系网中的所有顶点和表示层次关系的弧抽取出来，可得一个子图。由于父关系和子关系是同时出现的，即存在一条父关系就会同时存在一条与之对应的子关系，所以将两条弧合并为一条无向边，可以证明这个新的子图是一颗树，或者为森林，且每棵树均符合知识树的定义。这样就基于层次关系从知识点多关系网中得到了知识树，从而表示知识点间的父子、兄弟关系。

3.2.2 依赖关系

知识点在学习过程中具有一种先后关系。一个知识点当前是否可以学习往往取决于另一些知识点是否学习过，或者说后者是前者的预备知识。也就是说，一些知识点是依赖与某些知识点的。前驱关系、后继关系称作知识点之间的依赖关系，它定义了知识的学习顺序。知识点 A 与知识点 B 是**前驱关系** (Previous, 简作 Prev.)，则必须要学习知识点 A 后才能够学习知识点 B; 知识点 B 与知识点 A 是**后继关系** (Posterior, 简作 Post.)，则学习知识点 A 后可以学习知识点 B。两者的关系可以用如图 3 表示。

$$A \xrightarrow{Prev.} B \quad \text{前驱关系}$$

$$B \xrightarrow{Post.} A \quad \text{后继关系}$$

图 3 前驱关系与后继关系示意图

将知识点看作是有向图的顶点，知识点之间前驱后继的这种依赖关系作为弧，可以得到一个表示知识点的前驱后继关系的 AOV 网。可以证明，AOV 网是知识点多关系图的一个子图，且如果知识点的前驱后继顺序编排合理，则不会存在回路。这个 AOV 网的拓扑序列即为知识点的先修后续顺序。

3.2.3 相关关系

严格来说，相关关系不能作为一种独立的知识点之间的关系，它一般是建立在上面对两类关系构建出来的树/森林以及 AOV 网的基础上得到的。所谓知识点的**相关关系** (Relation, 简作 Rel.)，可以认为是两个知识点在知识点多关系图中的相近程度或相似程度，通常用于在基于本文分析的这一数据结构的系统的推荐算法中。为了简化知识点多关系图中知识点的逻辑关系，不赋予其独立的语义关系。

通过查阅文献，现总结两种方法来获取知识点间的相关关系权重：一种为根据两知识点的深度平均值和其最近父知识点的深度来计算得到，即公式(1)^[7]；另一种为将知识点的关系表示为低维空间中的向量，计算其相似度（欧氏距离）得到，即公式(2)^[8]。这样就可以通过设置阈值的方法来判定是否为相关关系，从而达到简化知识点间关系的表示的目的，进而简化知识点多关系图。

$$W(A, B) = \frac{\text{depth}(\text{parent}(A, B))}{\text{distance}(A, B)} = \frac{\text{depth}(\text{parent}(A, B))}{\sqrt{\frac{(\text{depth}(A))^2 + (\text{depth}(B))^2}{2}}} \quad (1)$$

其中 parent 表示最近共同父知识点， depth 表示最近共同父结点到树根路径长度， distance 表示结点到树根结点的平均距离。权重值 $W(A, B)$ 越大代表这两者关联越强，否则两者关联越弱。

$$\sin_{sem}(A, B) = 1 - \frac{\|A - B\|}{\|A - B\| + 1} \quad (2)$$

当结果 $\sin_{sem}(A, B)$ 数值越接近 1，两个向量越相似，两者语义相似度越高；越接近 0，两者语义相似度越低。

3.3 知识点多关系图的形式化定义

综上所述，知识点多关系图这一数据结构可定义如下。

知识点多关系图中的数据元素为知识点，称为**顶点**。知识点之间的关系为一个三元组 (H, R, T) ，称作知识点多关系图的**弧**（或有向边），其中 H ， T 分别为主体知识点和客体知识点， R 为主体知识点 H 到客体知识点 T 的关系，为子关系（Child）、父关系（Parent）、前驱关系（Prev.）、后继关系（Post.）中的一种。

知识点多关系图是一个有向图 $K = (V, E)$ 。其中 V 是顶点集，即知识点的集合； E 是顶点间弧的集合，即知识点间关系三元组 (H, R, T) 的集合。

3.4 模块、课程的组成

模块是知识点的集合，一个模块中通常包含许多知识点，而这些知识点之间往往是有依赖关系的。有时，一个模块本身就是一个知识点，层次结构上从属于它的知识点即可作为它的组成。由此，模块可以表示为知识点的集合，且带有每个知识的关系。一个模块中知识点的安排顺序可以通过组成模块的知识点的拓扑序列给出。同时，如果多个模块包含之间知识点之间存在前驱后继的依赖关系，那么这种关系可以转变为模块之间的依赖关系。也就是说，模块可以看作是更高一层次的知识点。

众多模块组成了一个课程，课程可以定义为模块的有序表，其中各个模块的顺序由模块之间的依赖关系给出。这样，作为最高层级的课程也就有了组织。一个个性化学习平台上的多个课程之间的依赖关系也可以类似由模块之间的依赖关系得到，从而转变为课程的依赖关系。至此，我们就将课程、模块、知识点统一为相似的结构，可以采用统一的数据结构来表示和处理，并且从课程到模块、模块到知识点是一种具有层次关系的树形关系。

4 存储结构设计

为了在计算机中表示知识点多关系图，下面讨论其在计算机中的映像。

4.1 基于关系三元组表的存储结构

由 3.3 节给定的知识点多关系图的定义，很容易想到一种基于顺序存储结构来存储

知识点多关系图的方法。这种存储方法与线性表中静态链表的思想相似，对于顶点集和有向边集均采用顺序存储的方式。

使用一个数组来存储知识点，并以每个知识点在数组中的下标索引作为其唯一标识符，如图 4 所示。使用一个顺序存储的三元组表（一般为三元组的一维数组）来存储知识点间的关系三元组，其中主体知识点和客体知识点存储的均为知识点的唯一标识符，即在知识点表中的下标索引，标志域 **relation** 表示两知识点间关系的种类，如图 5 所示。

index	0	1	2	3	4	5	6	7
knowledge	A	B	C	D	E	F	G	H

图 4 知识点集

subject knowledge	relation	object knowledge
0	Parent	1
0	Prev.	2
1	Child	0
2	Post.	0
⋮	⋮	⋮

图 5 关系三元组表

下面给出基于关系三元组表的知识点多关系图的类型定义：

```
class KnowledgeRelationalGraph {
private:
    struct RelationTuple {
        int subject;
        Relation relation;
        int object;
    };
    KnowledgePoint knowledge[Size];
    RelationTuple relationTuples[Size];
    // public methods...
};
```

这种存储方式是知识点多关系图最为直观的一种存储方式，也是这种存储结构的优点所在。通过一次遍历，可以得到多关系图中蕴涵的表示知识点的层级关系的知识树（森林）以及表示知识点的前驱后继关系的 AOV 网，可以使用相关的算法来得到知识点的拓扑序列、相关关系等。

但是这种存储方法也存在问题。当知识点集的规模和关系的复杂程度逐渐增大后，关系三元组表的规模将呈指数级别生长，会造成存储空间的大量占用。所以，对于知识点集的规模不太大，关系不太复杂的知识点多关系图可以采用这种存储方法。

4.2 基于邻接表的存储结构

根据知识点多关系图的定义，它也是一种有向图，故用来存储图的邻接表存储结构也可以应用到知识点多关系图的存储上来。知识点多关系图与普通的有向图相比，不同之处主要在于弧有了几种特殊的含义，故对邻接表稍作修改即可用它来存储知识点多关系图。

邻接表是一种顺序存储与链式存储相结合的方法，对于图中的每个顶点，将其所有的邻接点链接成一个单链表，称为该顶点的出边表，为了便于对所有出边表的表头指针进行存取操作，可以采取顺序存储的方式。存储出边表头指针的数组和存储顶点的数组构成了邻接表的表头数组，称为顶点表，这里依旧称为知识点表。因此，在邻接表中存在两种结点结构：顶点结点和出边表结点，如图 6 所示。

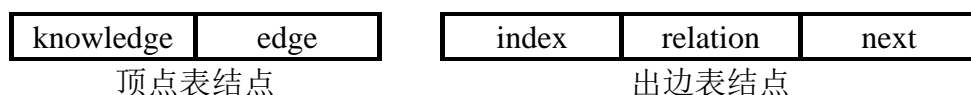


图 6 邻接表的结点结构

其中 knowledge 为知识点（主体）域，edge 为指针域，指向第一条出边；index 为客体知识点域，存放相应知识点在顶点表中的下标索引（注意为了避免重复存储知识点数据，index 域不存储知识点的数据信息）；relation 为标志域，表示两知识点间关系的种类；next 为指针域，指向出边表的下一个结点，即与主体知识点关联的下一个客体知识点。下面给出基于邻接表的知识点多关系图的类型定义：

```
class KnowledgeRelationalGraph {
private:
    struct EdgeNode {
        int index;
        Relation relation;
        EdgeNode* next;
    };
    struct KnowledgeNode{
        KnowledgePoint knowledge;
        EdgeNode* edge;
    };
    KnowledgeNode knowledge[Size];
    int knowledgeCount, relationCount; // 知识点数和关系数
    // public methods...
};
```

基于这种存储结构，可以将知识点多关系图以一种类似于图的存储方法存储下来，使得对图已有的算法稍作修改就可以应用于知识点多关系图。

5 反思与总结

本文从个性化学习应用中知识点的关系和模型入手,参考知识图谱的理论,给出了表征知识点、模块、课程在现实中关系的数据结构——知识点多关系图,并分析了其逻辑结构以及存储结构。文中提到的知识点在本文中探讨的数据结构中是不可再分的,而在实际应用中可以根据需求进行扩充,以完善其的属性,从而从简到繁,逐渐完善知识点的结构。

文中针对知识点的逻辑关系,提出了三大类关系:层次关系、依赖关系、相关关系,并给出了相应的定义。实际的应用中可以在此基础上扩充或精简,来丰富知识点间的语义关系或简化多关系图。例如,文中提到的两种关系,其下的关系本质上是成对出现的,即这些关系是双向的。若在不产生歧义的前提下,只考虑一个方向上的关系,将两种合并为一种,则可以简化知识点多关系图。

知识点多关系图这一数据结构,本质上可以看作有向图的一种扩充,即为每条弧赋予不同的类型,从而使得每条边可以表示知识点之间不同的关系,构建出语义网络。在实际应用中这一数据结构可以作为知识表示中的知识体,通过其它方面的设计和完善来支撑相关的人工智能应用。

参 考 文 献

- [1] 陈世福,陈兆乾,等.人工智能与知识工程[M].南京大学出版社:南京,1997.12:68-70.
- [2] 贾巍,瞿堃.知识表示视野下网络课程知识点关系研究[J].继续教育研究,2008,(5):62-64.
- [3] 王冬青,殷红岩.基于知识图谱的个性化习题推荐系统设计研究[J].中国教育信息化,2019,(17):81-86.
- [4] 刘峤,李杨,段宏,等.知识图谱构建技术综述[J].计算机研究与发展,2016,53(3):582-600.
- [5] Wang Z, Zhang J, Feng J, *et al.* Knowledge graph embedding by translating on hyperplanes[C]//AAAI. 2014, 14(2014): 1112-1119.
- [6] 田莉霞.知识图谱研究综述[J].软件,2020,41(4):67-71.
- [7] 蒋昌猛,冯筠,孙霞,等.基于知识点层次图的个性化习题推荐算法[J].计算机工程与应用,2018,54(10):229-235.
- [8] 郝卫,魏赞.基于知识图谱表示学习的推荐算法优化[J].智能计算机与应用,2020,10(4):22-26,31.