

# 本体稀疏矩阵学习以及在相似度计算中的应用

兰美辉<sup>1</sup>, 范全润<sup>1</sup>, 高 炜<sup>2</sup>

1. 曲靖师范学院 信息工程学院, 云南 曲靖 655011; 2. 云南师范大学 信息学院, 昆明 650500

**摘要:** 在大数据背景下, 本体所包含的概念越来越多, 其结构也越来越复杂. 这要求其对应的本体算法能高效地降低计算的维度, 进而减少计算复杂度. 将原有的本体稀疏向量学习模型进行扩展, 提出本体稀疏矩阵学习模型. 通过矩阵导数计算设计一种迭代算法来获取逼近最优解. 实验表明新算法在特定的本体应用领域有较高的效率.

**关键词:** 本体; 相似度计算; 本体映射; 稀疏矩阵

**中图分类号:** TP391

**文献标志码:** A

**文章编号:** 1673-9868(2020)01-0118-06

本体作为一种结构化数据存储、表示、计算的模型, 越来越受到广大研究者的重视<sup>[1-4]</sup>. 首先, 作为结构化模型, 在本体中数据的存放不是单纯的记录形式, 而是图的形式结构化存储数据, 图中的边表示数据之间的内在联系. 其次, 事实证明通过图模型并利用统计和图论的知识, 对处理本体中的数据信息有一定的优势.

最近, 针对本体的特殊框架和应用背景, 涌现了诸多本体学习算法<sup>[5-14]</sup>, 其中稀疏向量学习算法被广泛关注并应用于本体学习中. 基于稀疏向量学习的本体算法通过本体稀疏向量对高维本体顶点向量进行有效信息提取, 获取最有价值的信息, 并将每个本体顶点映射成实数.

本文提出一种扩展的本体稀疏向量计算方法, 并将该算法应用于两个特殊的工程领域来验证算法的有效性.

## 1 基于稀疏矩阵学习的本体算法描述

在本体图建模后, 需要将每个顶点对应概念的信息用一个统一维度的向量来表示. 设  $\mathbf{v} = (v^1, \dots, v^p)$  是顶点对应的  $p$  维向量.

本体函数通过本体稀疏向量可表示为

$$y = \sum_{i=1}^p v^i \beta^i + \beta_0 = \mathbf{v}\boldsymbol{\beta}^T + \beta_0 \quad (1)$$

其中:  $\boldsymbol{\beta} = (\beta^1, \dots, \beta^p)$  表示本体稀疏向量, 其大部分分量均为 0;  $\beta_0$  是一个表示误差的项. 经典本体稀疏向量  $\boldsymbol{\beta}$  学习模型为:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} l(\boldsymbol{\beta}) + Q(\boldsymbol{\beta}) \quad (2)$$

其中:  $l(\boldsymbol{\beta})$  为亏损项,  $Q(\boldsymbol{\beta})$  为控制本体稀疏向量  $\boldsymbol{\beta}$  稀疏度的项.

设  $\{(\mathbf{v}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \mathbb{R}$  为本体训练样本, 其中  $\mathbf{v}_i$  和  $y_i$  分别表示输入和输出. 设关系矩阵  $\mathbf{W} \in \mathbb{R}^{n \times n}$ , 其系数  $[\mathbf{W}]_{i,j} = r_{ij} \geq 0$  表示本体概念  $(\mathbf{v}_i, y_i)$  和  $(\mathbf{v}_j, y_j)$  之间的语义关系, 且有  $\mathbf{W} = \mathbf{W}^T$  和  $r_{11} = r_{22} = \dots =$

$r_m = 0$  (对角线上元素均为0) 成立. 本文考虑由扩展的稀疏向量来得到本体函数的计算模型如下:

$$y_i = \sum_{j=1}^p v_i^j \beta_i^j + e_i = \mathbf{v}_i \boldsymbol{\beta}_i^T + e_i \quad (3)$$

其中:  $\boldsymbol{\beta}_i = (\beta_i^1, \dots, \beta_i^p) \in \mathbb{R}^p$ , 偏移量  $e_i \in \mathbb{R}$  服从  $N(0, \sigma^2)$  分布. 对于每个  $\mathbf{v}_i$ , 计算其对应的  $y_i$ , 式(3)和式(1)的差别在于以下两点:

- 1) 式(1)中对于所有  $\mathbf{v}_i$ , 所求内积都是用同一个稀疏向量  $\boldsymbol{\beta}$ , 而式(3)中不同的  $\mathbf{v}_i$  对应不同的稀疏向量  $\boldsymbol{\beta}_i$ ;
- 2) 式(1)中对于所有  $\mathbf{v}_i$ , 其误差项都是相同的, 而在式(3)中, 不同的  $\mathbf{v}_i$  对应不同的误差  $e_i$ . 由此可知, 计算模型(3)是计算模型(1)的推广, 当  $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \dots = \boldsymbol{\beta}_n = \boldsymbol{\beta}$  且  $e_1 = e_2 = \dots = e_n = \beta_0$  时, 式(3)退化为式(1).

通过计算式(3), 我们要学习的不止是单个稀疏向量  $\boldsymbol{\beta}$ , 而是一组稀疏向量:  $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_n$ . 将这组稀疏向量进行合并得到稀疏矩阵  $\boldsymbol{\Omega} = [\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \dots, \boldsymbol{\beta}_n^T] \in \mathbb{R}^{n \times p}$ , 进而学习的目标从本体稀疏向量学习转化为本体稀疏矩阵的学习. 本文考虑的本体学习算法可以表示为:

$$\boldsymbol{\Omega} = \underset{\boldsymbol{\Omega}}{\operatorname{argmin}} J(\boldsymbol{\Omega}) \quad (4)$$

$$J(\boldsymbol{\Omega}) = \sum_{i=1}^n (y_i - \mathbf{v}_i \boldsymbol{\beta}_i^T)^2 + \Lambda(\boldsymbol{\Omega}, \mathbf{W}, \lambda_1, \lambda_2) \quad (5)$$

其中:  $\sum_{i=1}^n (y_i - \mathbf{v}_i \boldsymbol{\beta}_i^T)^2$  用来表示误差项,  $\Lambda(\boldsymbol{\Omega}, \mathbf{W}, \lambda_1, \lambda_2)$  用来控制本体稀疏矩阵  $\boldsymbol{\Omega}$  的稀疏程度. 在本体工程中, 一种常见的  $\Lambda(\boldsymbol{\Omega}, \mathbf{W}, \lambda_1, \lambda_2)$  设置方法如下:

$$\Lambda(\boldsymbol{\Omega}, \mathbf{W}, \lambda_1, \lambda_2) = \lambda_1 \sum_{i,j=1}^n r_{ij} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2 + \lambda_2 \sum_{i=1}^n \|\boldsymbol{\beta}_i\|_1^2$$

其中  $\lambda_1$  和  $\lambda_2$  为平衡调节参数. 当  $\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2$  的值比较小时, 可以认为  $\mathbf{v}_i$  和  $\mathbf{v}_j$  属于本体图的同一个团, 或者从数据的角度看属于同一个聚类中. 此时,  $r_{ij} > 0$ . 此外, 易知(5)式是凸的且存在全局最优解.

设  $\mathbf{V} = [\mathbf{v}_1^T, \dots, \mathbf{v}_n^T] = [\mathbf{u}_1, \dots, \mathbf{u}_p]^T$  为本体信息矩阵, 其中  $\mathbf{u}_i \in \mathbb{R}^n$ , (5)式可写成

$$J(\boldsymbol{\Omega}) = \|y - \boldsymbol{\Xi} \operatorname{vec}(\boldsymbol{\Omega})\|_2^2 + \Lambda(\boldsymbol{\Omega}, \mathbf{W}, \lambda_1, \lambda_2) \quad (6)$$

其中:  $\boldsymbol{\Xi} = [\operatorname{diag}(\mathbf{u}_1) \mid \operatorname{diag}(\mathbf{u}_2) \mid \dots \mid \operatorname{diag}(\mathbf{u}_p)] \in \mathbb{R}^{n \times (pn)}$ ,  $\operatorname{diag}(\mathbf{u}_i)$  表示对角线元素为  $\mathbf{u}_i$  中对应元素的对角矩阵,  $i \in \{1, \dots, p\}$ ; 向量化算子

$$\operatorname{vec}(\boldsymbol{\Omega}) = [[\boldsymbol{\Omega}]_{1,1}, [\boldsymbol{\Omega}]_{2,1}, \dots, [\boldsymbol{\Omega}]_{n,1}, \dots, [\boldsymbol{\Omega}]_{1,p}, [\boldsymbol{\Omega}]_{2,p}, \dots, [\boldsymbol{\Omega}]_{n,p}]^T \in \mathbb{R}^{pn}$$

设  $I_p \in \mathbb{R}^{p \times p}$  为单位矩阵,  $\otimes$  表示克罗内克乘积,  $I_{i,l} \in \{0, 1\}$  定义如下: 如果  $\operatorname{vec}(\boldsymbol{\Omega})$  的第  $l$  个元素  $[\operatorname{vec}(\boldsymbol{\Omega})]_l$  属于  $\boldsymbol{\beta}_i$  中的元素, 则  $I_{i,l} = 1$ ; 否则  $I_{i,l} = 0$ . 矩阵  $\mathbf{C}$  定义为:

$$[\mathbf{C}]_{i,j} = \begin{cases} \sum_{j'=1}^n \frac{r_{ij'}}{\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_{j'}\|_2} - \frac{r_{ij}}{\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2} & \text{若 } i=j \\ -\frac{r_{ij}}{\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2} & \text{若 } i \neq j \end{cases}$$

对角矩阵  $\mathbf{F}_e$  对角线上的元素为

$$[\mathbf{F}_e]_{l,l} = \sum_{i=1}^n \frac{I_{i,l} \|\boldsymbol{\beta}_i\|_1}{[\operatorname{vec}(\boldsymbol{\Omega})]_l}$$

令  $\mathbf{F}_g = I_p \otimes \mathbf{C}$ , 通过计算  $J(\boldsymbol{\Omega})$  关于  $\operatorname{vec}(\boldsymbol{\Omega})$  的导数, 可得

$$\operatorname{vec}(\boldsymbol{\Omega}) = (\boldsymbol{\Xi}^T \boldsymbol{\Xi} + \lambda_1 \mathbf{F}_g + \lambda_2 \mathbf{F}_e)^{-1} \boldsymbol{\Xi}^T y \quad (7)$$

由于(4)式为凸优化问题,  $\boldsymbol{\Omega}$  成为问题的全局最优解当且仅当其满足条件(7). 然而矩阵  $\mathbf{F}_g$  和  $\mathbf{F}_e$  依赖于  $\boldsymbol{\Omega}$ , 在  $\boldsymbol{\Omega}$  未知的情况下, 这两个矩阵无法计算. 因此, 通过优化如下目标函数来解本体问题(4):

$$\tilde{J}(\boldsymbol{\Omega}) = \|y - \boldsymbol{\Xi} \operatorname{vec}(\boldsymbol{\Omega})\|_2^2 + \operatorname{vec}(\boldsymbol{\Omega})^T (\lambda_1 \mathbf{F}_g^{(l)} + \lambda_2 \mathbf{F}_e^{(l)}) \operatorname{vec}(\boldsymbol{\Omega}) \quad (8)$$

其中  $\mathbf{F}_g^{(l)} = I_p \otimes \mathbf{C}^{(l)}$  是块对角矩阵,  $\mathbf{F}_e^{(l)} \in \mathbb{R}^{pn \times pn}$ .

$$[\mathbf{C}^{(t)}]_{i,j} = \begin{cases} \sum_{j'=1}^n \frac{r_{ij'}}{\|\boldsymbol{\beta}_i^{(t)} - \boldsymbol{\beta}_{j'}^{(t)}\|_2} - \frac{r_{ij}}{\|\boldsymbol{\beta}_i^{(t)} - \boldsymbol{\beta}_j^{(t)}\|_2} & \text{若 } i=j \\ -\frac{r_{ij}}{\|\boldsymbol{\beta}_i^{(t)} - \boldsymbol{\beta}_j^{(t)}\|_2} & \text{若 } i \neq j \end{cases}$$

$\mathbf{F}_e^{(t)} \in \mathbb{R}^{pn \times pn}$  是对角矩阵, 定义为

$$[\mathbf{F}_e^{(t)}]_{l,l} = \sum_{i=1}^n \frac{I_{i,l} \|\boldsymbol{\beta}_i^{(t)}\|_1}{[\text{vec}(\boldsymbol{\Omega}^{(t)})]_l}$$

用迭代平方最小策略可以得到最小化(8)式的逼近最优解: 给定  $\mathbf{F}_g^{(t)}$  和  $\mathbf{F}_e^{(t)}$ ,  $\boldsymbol{\Omega}$  的最优解可通过解

$$\frac{\partial \tilde{J}(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}} = 0$$

得到. 设

$$\mathbf{H}^{(t)} = \lambda_1 \mathbf{F}_g^{(t)} + \lambda_2 \mathbf{F}_e^{(t)}$$

可知

$$\text{vec}(\boldsymbol{\Omega}^{(t+1)}) = (\mathbf{H}^{(t)})^{-1} \boldsymbol{\Xi}^T (I_n + \boldsymbol{\Xi}(\mathbf{H}^{(t)})^{-1} \boldsymbol{\Xi}^T)^{-1} \mathbf{y} \quad (9)$$

得到  $\boldsymbol{\Omega}^{(t+1)}$  后, 再更新  $\mathbf{F}_g^{(t+1)}$  和  $\mathbf{F}_e^{(t+1)}$ .

整个本体学习算法概括起来描述如下.

**算法 A** 基于本体稀疏矩阵学习的本体相似度计算和本体映射算法

步骤 1: 输入本体图(对于本体映射, 则输入多本体图), 将每个本体概念对应顶点的所有语义信息用一个  $p$  维矩阵表示.

步骤 2: 确定本体样本集合  $\{(\mathbf{v}_i, \mathbf{y}_i)\}_{i=1}^n$ , 从而得到本体信息矩阵  $\mathbf{V}, \boldsymbol{\Xi}$ , 目标向量  $\mathbf{y}$ , 关系矩阵  $\mathbf{W}$  以及两个平衡参数  $\lambda_1$  和  $\lambda_2$ .

步骤 3: 初始化计数变量  $t=0$ , 并设置  $\mathbf{F}_g^{(0)}$  和  $\mathbf{F}_e^{(0)}$  的值;

步骤 4: 重复以下迭代直到收敛:

计算  $\text{vec}(\boldsymbol{\Omega}^{(t+1)}) = (\mathbf{H}^{(t)})^{-1} \boldsymbol{\Xi}^T (I_n + \boldsymbol{\Xi}(\mathbf{H}^{(t)})^{-1} \boldsymbol{\Xi}^T)^{-1} \mathbf{y}$ ;

更新  $\mathbf{F}_g^{(t+1)} = I_p \otimes \mathbf{C}^{(t+1)}$  以及  $[\mathbf{F}_e^{(t+1)}]_{l,l} = \sum_{i=1}^n \frac{I_{i,l} \|\boldsymbol{\beta}_i^{(t+1)}\|_1}{[\text{vec}(|\boldsymbol{\Omega}^{(t+1)}|)]_l}$ ;

$t=t+1$ .

步骤 5: 输出本体稀疏矩阵  $\boldsymbol{\Omega}$ , 并根据(3)式计算本体图中每个顶点对应的实数.

步骤 6: 通过两个顶点对应实数的一维距离来判断顶点对应本体概念之间的相似程度: 距离越短则相似度越高, 距离越远则相似度越小. 对于本体映射, 只计算不同本体概念之间的相似度.

步骤 7: 选择合理的策略, 给每个本体概念一个高相似度概念列表, 并返回给用户.

## 2 实 验

通过下面两个实验, 将上述算法分别应用到 PO 植物学本体和仿生机器人本体, 进而分别验证新本体稀疏矩阵学习算法对特定应用领域本体相似度计算和本体映射构建的效率. 得到最优本体稀疏矩阵  $\boldsymbol{\Omega}$  后,

忽略噪声项  $e_1, e_2, \dots, e_n$ , 通过  $\mathbf{y}_i = \sum_{j=1}^p v_i^j \boldsymbol{\beta}_i^j$  来计算每个顶点对应的实数, 最后通过实数之间的差值大小来判定本体概念之间的相似度.

### 2.1 本体相似度计算实验

首先验证本文本体稀疏矩阵学习算法是否可应用于植物学 PO 本体  $O_1$  (其基本结构见图 1), 该本体相关数据来自 <http://www.plantontology.org>, 其本质是一个给植物学领域专家使用的字典, 也可以看成一个有查找和分析功能的数据库. 最后数据的效率使用传统的  $P@N$ <sup>[15]</sup> 平均准确率来衡量, 同时在  $N=3, 5, 10$  时对这 4 类准确率进行对比. 将本体多重分割框架下的无穷推进算法<sup>[16]</sup>、本体多重分割框架下基于特殊亏损函数的本体学习算法<sup>[17]</sup>和基于梯度计算的本体稀疏向量学习算法<sup>[18]</sup>作用于植物学 PO 本体, 实验结果见表 1.

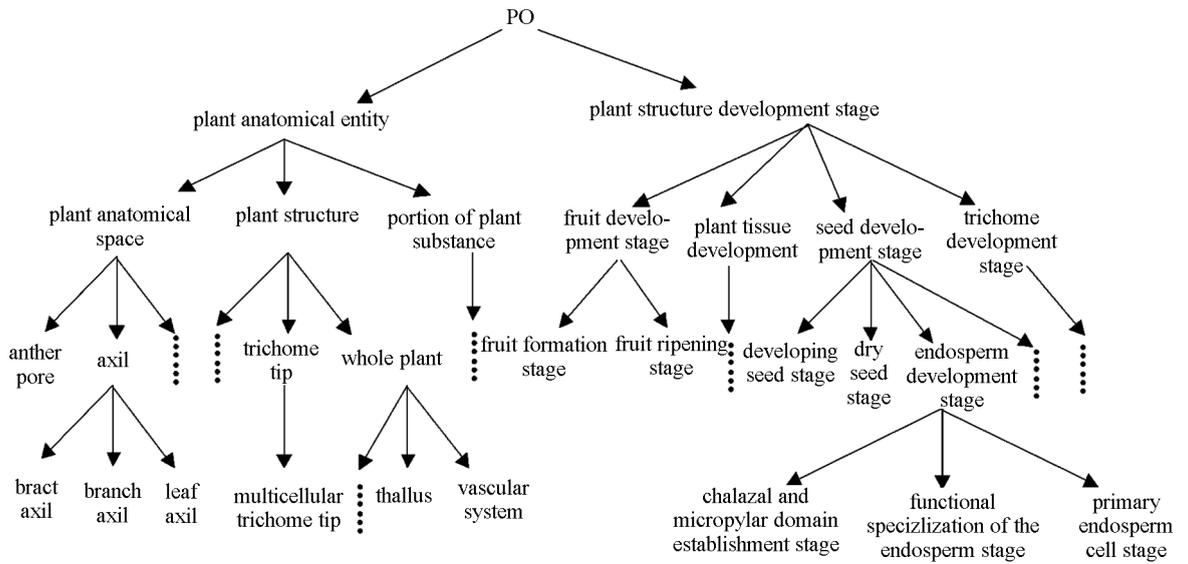


图 1 PO 本体  $O_1$

表 1 实验 1 部分数据

算法名称	$P@3$ 平均准确率	$P@5$ 平均准确率	$P@10$ 平均准确率
本文算法	0.523 8	0.636 8	0.841 3
无限推进算法 <sup>[16]</sup>	0.510 3	0.617 6	0.790 4
特殊亏损函数算法 <sup>[17]</sup>	0.488 5	0.570 2	0.726 3
梯度计算本体稀疏向量学习算法 <sup>[18]</sup>	0.486 4	0.579 6	0.708 2

由表 1 可知, 本文本体稀疏矩阵学习算法准确率要明显高于其他 3 类本体学习算法。

## 2.2 本体映射实验

第二个实验是将本文中本体稀疏矩阵学习算法应用于仿生学领域的两个“仿生机器人”本体  $O_2$  和  $O_3$  (图 2,3), 通过在两个仿生学本体顶点之间的相似度计算来得到本体映射, 但实验数据只关注于两个本体之间的相似度计算. 两个本体由文献[19]构造, 用于表示机器人的各个部件的关联. 将本体多重分割框架下的无穷推进算法<sup>[16]</sup>、本体多重分割框架下基于特殊亏损函数的本体学习算法<sup>[17]</sup>和基于梯度计算的本体稀疏向量学习算法<sup>[18]</sup>也应用于“仿生机器人”本体  $O_2$  和  $O_3$ . 实验数据结果同样采用  $P@N$  准确率的形式来对比, 表 2 给出当  $N$  的取值为 1,3,5 时部分对比结果。

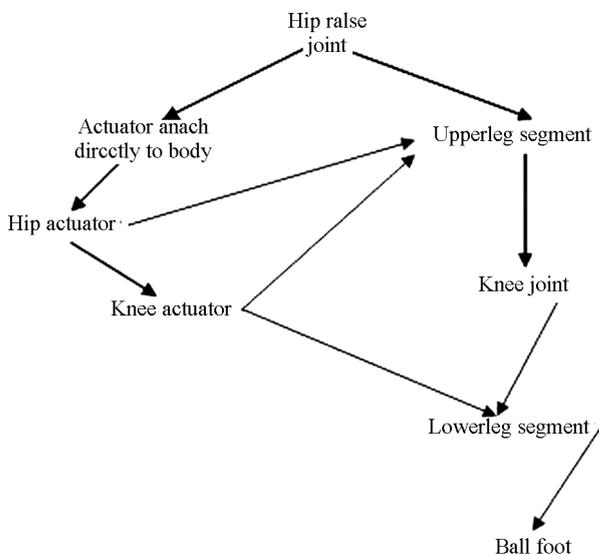


图 2 “仿生机器人”本体  $O_2$

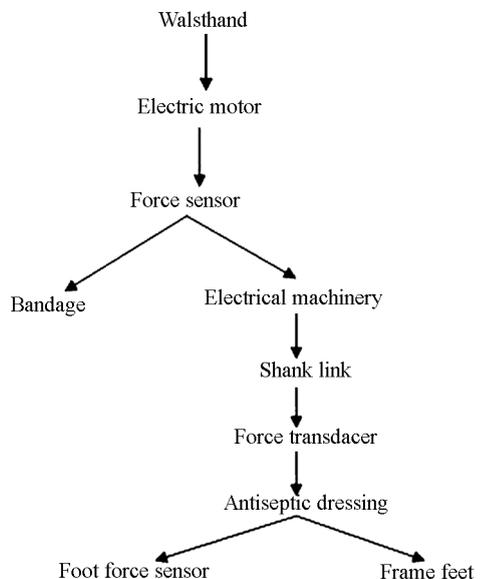


图 3 “仿生机器人”本体  $O_3$

表 2 实验 2 部分数据

算法名称	$P@1$ 平均准确率	$P@3$ 平均准确率	$P@5$ 平均准确率
本文算法	0.277 8	0.537 0	0.722 2
无限推进算法 <sup>[16]</sup>	0.277 8	0.500 0	0.688 9
特殊亏损函数算法 <sup>[17]</sup>	0.277 8	0.500 0	0.588 9
梯度计算本体稀疏向量学习算法 <sup>[18]</sup>	0.277 8	0.500 0	0.655 6

由表 2 中数据对比可知,本文所提出的本体稀疏矩阵学习算法在“仿生机器人”本体  $O_2$  和  $O_3$  间进行相似度计算的效率要明显高于其他 3 类本体学习算法,且随着  $N$  的增大,这种优势明显增加。

### 3 结束语

本文主要研究本体稀疏向量学习在本体相似度计算和本体映射中的应用。与以往文章的差别在于,考虑给每个概念一个特定的本体稀疏向量,进而整个算法归结于学习一个本体稀疏矩阵。从矩阵的导数计算出发,得到一个迭代策略来计算优化模型的逼近解。

#### 参考文献:

- [1] 兰美辉,任友俊,徐 坚,等.  $k$ -部排序本体相似度计算 [J]. 计算机应用, 2012, 32(4): 1094-1096.
- [2] 兰美辉,甘健侯,任友俊,等.  $k$ -部排序学习算法的可学习性分析 [J]. 西南大学学报(自然科学版), 2016, 38(3): 177-183.
- [3] 张太华,顾新建,何二宝. 产品知识模块本体的评价指标体系 [J]. 贵州师范大学学报(自然科学版), 2012, 30(1): 94-99.
- [4] 张 鹏,王国胤,陶春梅,等. 基于本体粗糙集的程序代码相似度度量方法 [J]. 重庆邮电大学学报(自然科学版), 2008, 20(6): 737-741.
- [5] GAO W, BAIG A Q, ALI H, et al. Margin Based Ontology Sparse Vector Learning Algorithm and Applied in Biology Science [J]. Saudi Journal of Biological Sciences, 2017, 24(1): 132-138.
- [6] GAO W, GUO Y, WANG K. Y. Ontology Algorithm Using Singular Value Decomposition and Applied in Multidisciplinary [J]. Cluster Computing-The Journal of Networks Software Tools and Applications, 2016, 19(4): 2201-2210.
- [7] GAO W, ZHU L L, WANG K Y. Ranking Based Ontology Scheming Using Eigenpair Computation [J]. Journal of Intelligent & Fuzzy Systems, 2016, 31(4): 2411-2419.
- [8] 吴剑章,朱林立,高 炜. 本体算法中相似度矩阵的学习 [J]. 小型微型计算机系统, 2015, 36(4): 773-777.
- [9] YAN L, LI Y J, YANG X, et al. Gradient Descent Technology for Sparse Vector Learning in Ontology Algorithms [J]. Journal of Discrete Mathematical Sciences & Cryptography, 2016, 19(3): 753-775.
- [10] WU J Z, YU X, GAO W. Similarity Matrix Learning for Ontology Application [J]. International Journal of Information Technology and Management, 2016, 15(1): 1-13.
- [11] 高 炜,梁 立,徐天伟. 基于正则化瑞利系数的半监督  $k$ -部排序学习算法及应用 [J]. 西南师范大学学报(自然科学版), 2014, 39(4): 124-128.
- [12] 高 炜,朱林立,梁 立. 基于图正则化模型的本体映射算法 [J]. 西南大学学报(自然科学版), 2012, 34(3): 118-121.
- [13] 朱林立,戴国洪,高 炜. 成对排序本体学习算法 [J]. 西南师范大学学报(自然科学版), 2013, 38(12): 101-106.
- [14] 吴剑章,余 晓,高 炜. 基于 Mahalanobis 矩阵学习的本体算法 [J]. 西南大学学报(自然科学版), 2015, 37(2): 117-122.
- [15] CRASWELL N, HAWKING D. Overview of the TREC 2003 Web Track [C]//Proceedings of the Twelfth Text Retrieval Conference. Maryland: NIST Special Publication, 2003: 78-92.
- [16] GAO W, ZHU L L, GUO Y. Multi-Dividing Infinite Push Ontology Algorithm [J]. Engineering Letters, 2015, 23(3): 132-139.

- [17] GAO W, LIANG L, XU T W. New Multi-Dividing Ontology Learning Algorithm Using Special Loss Functions [J]. *The Open Cybernetics & Systemics Journal*, 2014(8): 259-268.
- [18] GAO W, WU J Z, ZHU L L. Ontology Optimization Strategies for Sparse Vector Learning Using Gradient Descent Tricks [J]. *Journal of Computational Information Systems*, 2015, 11(17): 6393-6402.
- [19] GAO W, ZHU L L. Gradient Learning Algorithms for Ontology Computing [J]. *Computational Intelligence and Neuroscience*, 2014, 2014: 1-12.

## Ontology Sparse Matrix Learning and Its Application in Similarity Computation

LAN Mei-hui<sup>1</sup>, FAN Quan-run<sup>1</sup>, GAO Wei<sup>2</sup>

1. *School of Information Engineering, Qujing Normal University, Qujing Yunnan 655011, China;*

2. *School of Information, Yunnan Normal University, Kunming 650500, China*

**Abstract:** Under the background of big data, ontology contains more and more concepts, and thus its structure becomes more complex. Therefore, it is required that the corresponding ontology algorithm be able to reduce the computational dimension efficiently, so as to reduce the computation complexity. In this paper, the original ontology sparse vector learning model is extended, and an ontology sparse matrix learning model is proposed to obtain the optimal approximation solution. An iterative algorithm is designed to get this solution by means of matrix derivative computation. Two experiments verify that the new algorithm has higher efficiency in specific ontology specific applications.

**Key words:** ontology; similarity measure; ontology mapping; sparse matrix

责任编辑 张 桢