·人工智能及识别技术· 文章编号:1000-3428(2017)03-0204-09 文献标志码:A 中图分类号:TP301.6

一种数学表达式检索结果相关排序算法

田学东[1a],张凯歌[1a],周 南[1a],张植明[1b],田冰洁[2]

(1.河北大学 a.计算机科学与技术学院;b.数学与信息科学学院,河北 保定 071002;
2.河北金融学院 经济贸易系,河北 保定 071051)

摘 要: 针对数学表达式符号种类繁多、结构复杂多变、语法语义丰富等特点,提出一种检索结果相关排序算法,利用犹豫模糊集在处理多特征、多隶属度模式方面的优势,计算数学表达式间的相似度,实现基于相似度的数学表达式检索结果的相关排序。通过归纳数学表达式的符号、结构、语法、语义方面的特征,建立数学表达式的相似度函数,对数学表达式检索系统中用户查询式与检索结果集中数学表达式之间的相似程度进行综合多视角的测量。实验结果表明,该算法能实现数学表达式检索系统结果数据的有序输出,有助于改善数学表达式检索系统的性能。

关键词: 数学表达式;犹豫模糊集;检索;相似度;相关排序

中文引用格式: 田学东,张凯歌,周 南,等.一种数学表达式检索结果相关排序算法[J].计算机工程,2017,43(3):204-212.

英文引用格式: Tian Xuedong, Zhang Kaige, Zhou Nan, et al. A Relevance Ranking Algorithm of Mathematical Expression Retrieval Results[J]. Computer Engineering, 2017, 43(3):204-212.

A Relevance Ranking Algorithm of Mathematical Expression Retrieval Results

TIAN Xuedong[1a], ZHANG Kaige[1a], ZHOU Nan[1a], ZHANG Zhiming[1b], TIAN Bingjie[2]

(1a. College of Computer Science and Technology; 1b. College of Mathematics and Information Science, Hebei University, Baoding, Hebei 071002, China;
2. Department of Economic Trade, Hebei Finance University, Baoding, Hebei 071051, China)

【Abstract】 Aiming at the characteristics of mathematical expressions, such as the diversity of mathematical symbols, the complexity of structures and the richness of semantics, a mathematical expression retrieval results ranking algorithm is proposed. It utilizes the advantages of the hesitant fuzzy sets in dealing with the problems with multiple features and membership values to calculate the similarity between mathematical expressions, based on which the mathematical expressions are ranked. It measures the similarity of mathematical expressions through constructing the hesitant fuzzy membership with the multidimensional characteristics of the symbols, structures, grammar and semantics of mathematical expressions. Experimental results show that the algorithm can not only realize ordered output of search result data, but also improve the performance of mathematical expression retrieval system.

【Key words】 mathematical expression; hesitant fuzzy set; retrieval; similarity; relevance ranking

DOI: 10.3969/j.issn.1000-3428.2017.03.035

0 概述

数学表达式是科技文献的重要组成部分,随着信息化和科技文献数据库建设的飞速发展,以数学表达式为线索获取网络科技信息的需求与日俱增。然而,数学表达式不同于普通文本,属于多符号集构成的复杂二维模式,目前主流搜索引擎尚无法提供功能完善的数学检索服务,因此,数学表达式检索成为信息检索领域的研究热点。

目前,国内外相关研究机构已经对数学表达式检索进行了一些研究,提出的方法或原型系统包括DLMF(Digital Library of Mathematical Functions)

基金项目: 国家自然科学基金“数学表达式资源获取与检索模型研究”(61375075);保定市科学技术研究与发展指导计划项目(15ZR063)。

作者简介: 田学东(1963—),男,教授、博士,主研方向为信息检索、模式识别;张凯歌、周 南,硕士研究生;张植明,讲师、硕士;田冰洁,助教、硕士。

收稿日期: 2016-07-12 **修回日期:** 2016-08-15 **E-mail:** xuedong_tian@126.com

Search[1-2], MathDex[3], MathWebSearch[4-6], LeActive-Math[7-9], EgoMath[10-13], Mathsearch[14-16], WikiMirs[17-18], MIaS[19]等。从实现机理方面可以将其分为 2 类:对全文搜索引擎进行数学扩充的检索系统和专门进行数学表达式检索的检索系统。从匹配模式方面可以分为精确匹配和非精确匹配。从检索粒度方面可以分为针对数学表达式进行检索和针对数学文档进行检索。

信息检索主要关注如何从存储的信息集合中快速获取需要的信息[20]。因此,在实现了基本的数学表达式检索功能后,如何将查询结果按照用户查询的真实需求进行有序输出,同样是实现数学检索系统的关键问题。但由于数学表达式复杂的二维结构及语法语义特征造成的数学表达式信息多样性,对数学表达式全面评价并进行排序仍存在困难。文献[21]提出的模糊集扩展形式——犹豫模糊集理论,能够有效避免传统模糊集在多特征分类问题中进行算子集结所导致的信息丢失问题,因此被广泛应用于群决策信息的处理问题中,为数学表达式相似评价带来了新的途径。

本文提出一种基于犹豫模糊集的数学表达式检索结果相关排序算法。利用犹豫模糊集在多特征模式评价方面的优势,在对数学表达式的符号、空间、语法、语义等方面特征进行归纳的基础上,建立融合数学表达式多维特征的犹豫隶属度函数,计算犹豫模糊集合之间的距离,实现对数学表达式查询式与检索结果集之间的相似度计算,进而完成对数学检索结果的相关排序。

1　相关理论与技术

1.1　数学表达式相似度

在用户输入待查询数学表达式后,数学表达式检索系统会返回大量与之相关的数学表达式。为了便于用户尽快获取所需要的数学内容,通常采用计算用户查询式和检索获取的数学表达式之间的相似度实现对检索结果的相关排序及输出。

现有的数学检索文献中涉及到数学表达式相似度计算的内容相对较少。文献[22]提出一种基于结构相似度的数学检索方法,采用“树编辑距离”计算 Presentation MathML 格式的数学表达式的相似度,并设计了 top-k 选择算法和索引算法来减少查询处理时间;在采集的 Wikipedia 和 DLMF 中含数学表达式的文档上,验证了方法的有效性。文献[23]将传统的面向文档排序的 TF-IDF 算法应用到数学表达式相似度计算中,将每一个表达式视为一个单独的文档建立索引,这一方法在进行数学表达式相似度计算时避免了由文本和相似公式导致的噪声。文献[17]同样采取改进的 TF-IDF 算法计算数学表达式相似度,将表达式转化为树并进行一般化,结合该关键词的层次与是否一般化等信息,计算得出表达式间的相似度。文献[18]在该方法的基础上,考虑查询表达式被匹配表达式匹配到的关键字数目与查询表达式中关键字总数的比例,对表达式相似度计算方法进行了改进。文献[24]用五元组(s,n,r,p,b)表示数学表达式,通过计算表达式所对应的五元组的距离作为相似度。文献[25]采用解析树表示数学表达式,通过定义函数分类距离、数据类型层级、匹配深度、查询覆盖度以及公式/表达式特征,利用对查询表达式和获取的目标表达式的解析树进行递归的相似距离分析,计算它们之间的相似度。在 DLMF 的样本集上验证了所提出数学表达式相似度的有效性。文献[26]利用二叉树表示数学表达式,进行归一化处理,并定义了相应的相似度计算公式。

上述文献对数学表达式的相似度评价进行了有益的尝试,但由于数学表达式在符号自身以及结构、语法、语义方面的灵活性和多样性,使得现有数学表达式相似度评价方法无论是在特征的全面性与多样性,还是在多特征融合的有效性与合理性方面,存在许多有待解决的问题。本文所采用的犹豫模糊集这一新的理论和方法,擅长处理多属性评价问题,能够有效避免传统模糊集在对多特征进行算子集结时所导致的信息丢失问题,恰好适用于解决数学表达式这一复杂二维模式的多属性决策问题,有助于提高数学表达式相似评价的客观性和准确性。

1.2　犹豫模糊集

定义 1　(犹豫模糊集)设 X 是一个非空集合,则称:

$$E=\{\langle x,h_E(x)\rangle \mid x\in X\} \tag{1}$$

为犹豫模糊集,其中 $h_E(x)$ 称为犹豫模糊元素,是元素 x 对于集合 E 的几个可能的隶属度的集合,其元素值在[0,1]上分布[21]。

犹豫模糊集中每个元素的隶属度不是一个确定的值或分布,而是若干可能的值。

文献[27]提出了一系列犹豫模糊集距离测度、相似性测度概念,设 A 和 B 分别为非空集合 $X=\{x_1,x_2,\cdots,x_n\}$ 上的犹豫模糊集合,A 和 B 的广义犹豫标准距离、相似度分别表示为:

$$d_{\mathrm{ghn}}(A,B)=\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{l_{x_i}}\sum_{j=1}^{l_{x_i}}\left|h_A^{\sigma(j)}(x_i)-h_B^{\sigma(j)}(x_i)\right|^{\lambda}\right)\right]^{1/\lambda} \tag{2}$$

$$s(A,B)=1-d_{\mathrm{ghn}}(A,B) \tag{3}$$

其中，$d_{\mathrm{ghn}}(A,B)$表示犹豫模糊集 A 和 B 的广义犹豫标准距离；$s(A,B)$为对应的相似度；λ 表示控制参数，当 $\lambda=1$ 时 $d_{\mathrm{ghn}}(A,B)$退化为犹豫模糊汉明距离，$\lambda=2$ 时退化为犹豫模糊欧几里德距离；$h_A^{\sigma(j)}(x_i)$和 $h_B^{\sigma(j)}(x_i)$分别表示犹豫模糊元素 $h_A(x_i)$和 $h_B(x_i)$中第 j 大的元素值，$l_{x_i}=\max(l_A(x_i),l_B(x_i))$，其中 $l_A(x_i)$和 $l_B(x_i)$表示 $h_A(x_i)$和 $h_B(x_i)$中元素的个数[27]。

基于犹豫模糊集理论，犹豫模糊信息熵、交叉熵和相似度测度[28]概念与公理化定义也被提出。其后，区间值犹豫模糊集[29]及其相应的一些关联度、距离及相似性测度、算子和相应的决策方法逐步被提出[30]；作为一项新的理论，其应用还有许多有待解决的问题。

当数学表达式检索系统依据查询表达式进行检索时，需要对每个结果表达式与查询表达式的相似程度进行评价以提供给用户经过相关排序的检索结果。相似评价需要考察表达式多方面的特征，普通模糊集在评价过程中只能针对某一方面给出一个评价值，影响了相似度计算的准确性。利用犹豫模糊集评价理论能对数学表达式在每个属性下给出多个可能的隶属度评价值，能对表达式间的相似程度进行更全面、更贴近实际情况的评价。

2 基于犹豫模糊集的数学表达式相似度

2.1 数学表达式相似度评价流程

基于犹豫模糊集的数学表达式相似度评价流程主要分为数学表达式包含匹配、表达式特征提取、计算表达式隶属度以及表达式相似度计算 4 个部分。

文献[31]提出的数学表达式检索系统包含 3 种匹配模式：精确匹配，包含匹配以及运算符匹配。其中，第 1 种精确匹配模式要求结果表达式与查询表达式在符号和结构上完全一致，不存在相似问题；第 3 种运算符匹配只对表达式中的运算符进行匹配而对符号及其运算结构没有要求，也没有相似度评价的必要；而第 2 种包含匹配模式要求查询表达式是结果表达式的子式，结果集合不仅包含用户查询成分还对此进行了扩充，非常类似于全文检索中的关键词匹配，包含的表达式信息更为全面，更适合也更需要进行相似度评价。因此，本文以包含匹配查询方式的结果作为相关排序对象。

设用户数学查询式为 F_{q}，检索结果集合中的任一元素（数学表达式）为 $F_{\mathrm{Rq}_i}(i=1,2,\cdots,n;n$ 为检索结果集合中数学表达式的总数），F_{Rq_i}均以 F_{q} 为子式，例如，当 $F_{\mathrm{q}}=a^2+2b$ 时，F_{Rq_i}可能是 $\sqrt{a^2+2b}+3$，$\dfrac{a^2+2b}{a+c}$或者 b^2+a^2+2b。F_{q} 和 F_{Rq_i}均以 LaTeX 形式描述，将其解析为表达式描述结构（Formula Description Structure，FDS）[31]，其中包含了表达式中每个符号的名称，在表达式中所处层次，是否为运算符，与上一层次符号的相对位置关系等特征。利用犹豫模糊集理论，选取 n 个评价属性$\{P_1,P_2,\cdots,P_n\}$，每个评价属性分别包含一组评价指标$\{Ind_{11},Ind_{12},\cdots,Ind_{1m_1}\}$，$\{Ind_{21},Ind_{22},\cdots,Ind_{2m_2}\}$，$\{Ind_{n1},Ind_{n2},\cdots,Ind_{nm_i}\}$对每个评价指标设置相应的隶属度函数$\{u_{11},u_{12},\cdots,u_{1m_1}\}$，$\{u_{21},u_{22},\cdots,u_{2m_2}\}$，$\{u_{n1},u_{n2},\cdots,u_{nm_i}\}$。将表达式的特征值代入相应的隶属度函数得到其对每个评价属性的多个隶属度，从而形成犹豫模糊集合。计算数学表达式间相似度即为计算其所对应的犹豫模糊集合间的相似度。基于犹豫模糊集的数学表达式评价原理如图 1 所示。

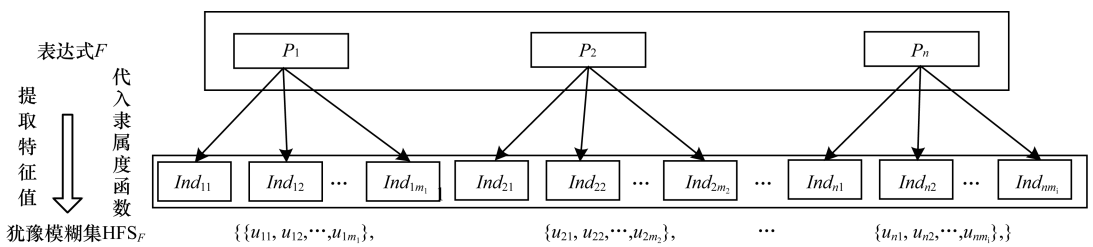

图 1 数学表达式评价原理

2.2 数学表达式的相似度评价属性

与普通文本不同，数学表达式在结构上为二维分布，在评价过程中不仅要考虑数学表达式字符信息的相似性，还要考虑数学表达式结构上的相似性。定义数学表达式评价属性为三元组$(P_{\mathrm{S}},P_{\mathrm{O}},P_{\mathrm{N}})$，其中，$P_{\mathrm{S}}$ 为表达式的结构属性；P_{O} 为表达式的运算符属性；P_{N} 为表达式的运算数属性，分别对表达式的结构、字符方面特征进行评价。每个评价属性中又包含若干评价指标，通过对每个指标设置隶属度函数，以评价查询表达式 F_{q} 以及每个结果表达式 F_{Rq_i}对于各个属性的犹豫隶属度。

1)结构属性 P_S

(1)Ind_L:层次指标。考察 F_q 在 F_{Rq_i} 中所处的层次,所处层次越接近主层次,即 0 层,表示 F_q 在 F_{Rq_i} 中越重要,则两者相似度越大。

设 $sim_{formulae}(F_{Rq_i},F_q)$ 表示表达式 F_q 与 F_{Rq_i} 的相似度,若 F_q 为 $a+b$,F_{Rq_1} 为 $\frac{a+b}{2}$,F_{Rq_2} 为 $a+b+2$,则表达式层次与相似度关系如图 2 所示。

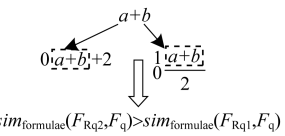

图 2　数学表达式层次与相似度关系

(2)Ind_N:长度指标。考察 F_{Rq_i} 与 F_q 的长度信息,即 F_{Rq_i} 与 F_q 包含的字符数目,F_q 与 F_{Rq_i} 包含的字符数目越接近,说明 F_q 与 F_{Rq_i} 的相似度越大。

若 F_q 为 $a+b$,F_{Rq_1} 为 $a+b+2$,F_{Rq_2} 为 $a+b+c+d+\sqrt{a^2+b^2}$,显然有:

$$sim_{formulae}(F_{Rq_2},F_q) < sim_{formulae}(F_{Rq_1},F_q) \quad (4)$$

(3)Ind_P:位置指标。考察 F_q 在 F_{Rq_i} 中的水平位置,即在表达式的 LaTeX 形式中,F_q 子式的初始位置是 F_{Rq_i} 中自左向右的第几个字符,F_q 在 F_{Rq_i} 中所处位置越靠前,说明 F_q 在 F_{Rq_i} 中的重要程度越高。

(4)Ind_F:标志位指标。考察 F_q 在 F_{Rq_i} 中与上一层次字符的相对位置。

若 F_q 为 $i+1$,F_{Rq_1} 为 x_{i+1},F_{Rq_2} 为 $\frac{i+1}{x}$,表达式标志位与相似度关系如图 3 所示。

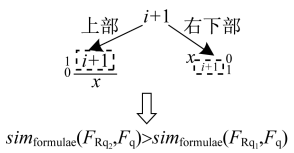

图 3　数学表达式标志位与相似度关系

2)运算符属性 P_O

设 F_q 中包含若干运算符 $\{O_1,O_2,\cdots,O_{l_1}\}$,其中 l_1 为 F_q 中包含的运算符个数,考察 F_{Rq_i} 中每个运算符 $O_m(m=1,2,\cdots,l_1)$ 出现的次数及其重要程度,作为评价指标 Ind_{O_m}。

3)运算数属性 P_N

设 F_q 中包含若干运算数 $\{N_1,N_2,\cdots,N_{l_2}\}$,其中 l_2 为 F_q 中包含的运算数个数,考察 F_{Rq_i} 中每个运算数 $N_n(n=1,2,\cdots,l_2)$ 出现的次数及其重要程度,作为一个评价指标 Ind_{N_n}。

2.3　数学表达式犹豫模糊隶属度的定义

犹豫模糊隶属度用于评价表达式对于各个评价指标的隶属程度,对于给定的查询表达式 F_q,计算 F_q 与所有结果表达式 F_{Rq_i} 对于每个评价指标的隶属度,对不同评价指标的隶属度函数进行如下定义:

定义 2　层次指标 Ind_L 的隶属度函数为:

$$u_{Ind_L}(F_q,F_{Rq_i}) = e^{-\alpha\cdot Ind_L} \quad (5)$$

其中,Ind_L 为子表达式 F_q 在 F_{Rq_i} 中所处的层次;α 为层次属性权重系数,通过统计数据库中数学表达式数据得到。

定义 3　长度指标 Ind_N 的隶属度函数为:

$$u_{Ind_N}(F_q,F_{Rq_i}) = \frac{length_{F_q}}{length_{F_{Rq_i}}} \quad (6)$$

其中,$length_{F_q}$ 表示 F_q 包含的字符数目;$length_{F_{Rq_i}}$ 表示 F_{Rq_i} 包含的字符数目。

定义 4　位置指标 Ind_P 的隶属度函数为:

$$u_{Ind_P}(F_q,F_{Rq_i}) = e^{-\beta\cdot(Ind_P-1)} \quad (7)$$

其中,Ind_P 为子表达式 F_q 在 F_{Rq_i} 中所处的水平位置;β 为位置属性权重系数,通过统计数据库中数学表达式数据得到。

定义 5　标志位指标 Ind_F 的隶属度函数为:

$$U_{Ind_F}(F_q,F_{Rq_i}) = \{(Ind_F,u_{Ind_F})\} \quad (8)$$

其中,Ind_F 为子表达式 F_q 在 F_{Rq_i} 中与上一层次字符的相对位置;u_{Ind_F} 为 Ind_F 对应的隶属度值。

定义 6　运算符指标 Ind_{O_m} 的隶属度函数为:

$$u_{Ind_{O_m}}(O_m,F_{Rq_i}) = iff_{O_m}\times\frac{count_{O_m}}{l_1} \quad (9)$$

其中,$iff_{O_m}=\frac{1}{\gamma}\lg\frac{C}{C_{O_m}}$,表示运算符 O_m 的权重;C_{O_m} 表示数据库中包含 O_m 的表达式个数;C 表示数据库中所有表达式的个数;γ 为符号权重系数,通过统计数据库中的表达式数据得出,$count_{O_m}$ 表示 F_{Rq_i} 中 O_m 出现的次数。

定义 7　运算数指标的隶属度函数为:

$$u_{Ind_{N_n}}(N_n,F_{Rq_i}) = iff_{N_n}\times\frac{count_{N_n}}{l_2} \quad (10)$$

其中,$iff_{N_n}=\frac{1}{\gamma}\lg\frac{C}{C_{N_n}}$,表示运算数 N_n 的权重;C_{N_n} 表示数据库中包含 N_n 的表达式个数;$count_{N_n}$ 表示 F_{Rq_i} 中 N_n 出现的次数。

对于表达式 F_q 在每个评价指标的隶属度,将式(5)~式(10)中的 F_{Rq_i} 替换为 F_q,进行相应计算。

2.4　数学表达式相似度计算

经过评价后,表达式 F_q,F_{Rq_i} 分别对应犹豫模糊集合 HFS_q 和 HFS_{Rq_i},任一评价属性为 $P_p(p=1,2,3)$,$h_{HFS_q}(P_p)$ 和 $h_{HFS_{Rq_i}}(P_p)$ 分别为犹豫模糊集合 HFS_q 和 HFS_{Rq_i} 的犹豫模糊元素,$h_{HFS_q}(P_p)$ 和 $h_{HFS_{Rq_i}}$

(P_p)中元素分别为表达式 F_q 和 F_{Rq_i}在属性 P_p 包含的各个评价指标下的隶属度值。将计算表达式相似度 $sim_{formulae}(F_q, F_{Rq_i})$转换为计算 2 个犹豫模糊集的相似度：

$$\begin{aligned} & sim_{formulae}(F_q, F_{Rq_i}) \\ & = s(HFS_q, HFS_{Rq_i}) \\ & = 1 - \left[\frac{1}{3}\sum_{p=1}^{3}\left(\frac{1}{l_{P_p}}\sum_{j=1}^{l_{P_p}}\left|h_{HFS_q}^{\sigma(j)}(P_p) - h_{HFS_{Rq_i}(P_p)|\lambda)}^{\sigma(j)}\right|\right)\right]^{1/\lambda} \end{aligned} \tag{11}$$

其中，l_{P_p}表示评价属性 P_p 中评价指标的个数；$h_{HFS_q}^{\sigma(j)}(P_p)$表示 F_q 在评价属性 P_p 中各个评价指标的隶属度中的第 $\sigma(j)$大的隶属度的值；$h_{HFS_{Rq_i}}^{\sigma(j)}(P_p)$表示 F_{Rq_i}在评价属性 P_p 中各个评价指标的隶属度中的第 $\sigma(j)$大的隶属度的值。

若表达式 F_{Rq_i}中包含多个与 F_q 相同的子式，则分别按照不同位置匹配到的子式进行相似度计算，取最大的相似度值作为最终表达式 F_{Rq_i}与 F_q 的相似度。

3 数学表达式相关排序算法

基于犹豫模糊集的数学表达式相关排序流程如图 4 所示。

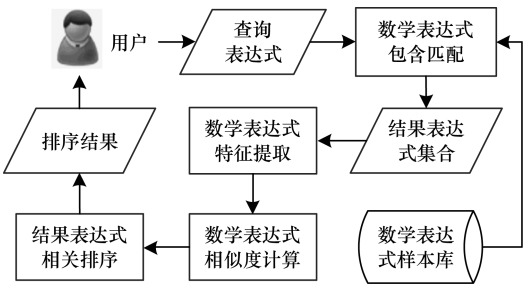

图 4 基于犹豫模糊集的数学表达式相关排序流程

算法步骤如下：

输入 LaTeX 形式的查询表达式

输出 LaTeX 形式结果表达式排序结果

1）初始化数据表 *NodeInfo*(*id*, *Expid*, *Level*, *Position*, *Flag*, *u_l*, *u_n*, *u_p*, *u_f*), *ExpInfo*(*Expid*, *Filename*, *Expstring*, *Len*), *OpInfo*(*Expid*, *nodeexp*, *weight*, *isoperator*, *u_o*), *Results*(*Expid*, *Expstring*, *sim*), *i*;//*id* 为查询表达式子式 *id*, *Expid* 为表达式 *id*, *Level* 为层次, *Length* 为长度, *Position* 为位置, *Flag* 为标志位, *u_l* 为层次隶属度, *u_n* 为长度隶属度, *u_p* 为位置隶属度, *u_f* 为标志位隶属度, *Expstring* 为表达式 *LaTeX* 字符串, *Filename* 为表达式文件名, *nodeexp* 为运算符/数字符串, *weight* 为运算符/数权重, *u_o* 为运算符/数隶属度, *isoperator* 表示该字符是否为运算符, *sim* 为表达式与查询表达式的相似度。

2）查询表达式 *Q_Id*、表达式字符串 *Q_Str*、文件名 *Q_Fil* 写入表 *ExpInfo*。

3）查询表达式解析，特征写入表 *NodeInfo*, *OptInfo*, *OpnInfo*，长度写入 *ExpInfo*。

4）*i* = *i* + 1，若 *i* > 数据库中表达式总数，执行 6)；否则，执行 5)。

5）选择查询表达式字符串 *Q_Str*，查询数据库中表达式表 *HASHTable*，若表达式 F_i 字符串包含 *Q_Str*，对该表达式解析，写入表 *ExpInfo*, *NodeInfo*, *OpInfo*，执行 4)。

6）根据 *NodeInfo* 信息计算隶属度，写入表 *NodeInfo*, *OpInfo*。

7）计算相似度，写入表 *Results*。

8）按 *sim* 降序，将 *Results* 表返回给用户。

经过对 138 539 条从网络上获取的数学表达式的统计，本实验所采用的隶属度函数参数值 α 为 1.468，β 为 0.66，γ 为 10，基于标志位属性的隶属度函数具体为：

$$\begin{aligned} U_{Ind_F}(F_q, F_{Rq_i}) &= \{(Ind_F, u_{Ind_F})\} \\ &= \{(0,1),(1,0.7),(2,0.55), \\ &\quad (4,0.3),(5,0.7),(6,0.75), \\ &\quad (7,0.25),(8,0.25)\} \end{aligned} \tag{12}$$

其中，层次属性权重系数 α 是通过统计数据库中所有表达式层次，确定最大层次、最小层次及层次分布中心，分别对 3 个特殊层次指定隶属度，并据此进行曲线拟合得到的；位置属性权重系数 β 是通过统计数据库中所有表达式长度信息经过相似步骤得到的；为确保每个符号权重不得大于 1，通过统计数据库中出现最少的符号 *c* 的频率，令符号权重系数 $\gamma = 10^M$，其中 *M* 为不超过 $\lg\frac{\text{所有表达式个数}}{\text{包含字符 } c \text{ 的表达式个数}}$的最大整数的位数；$(Ind_F, u_{Ind_F})$的确定是经过统计不同标志位所代表的典型运算，依据知识进行具体隶属度值的设定。

设 $F_q = a - b$, $F_{Rq_1} = a - b = c$, $F_{Rq_2} = (a-b)^2$, $F_{Rq_3} = c(a-b)$, $F_{Rq_4} = \frac{a+b}{a-b} = \frac{c+d}{c-d}$，对应的犹豫模糊集合如表 1 所示，相似度计算结果如表 2 所示。

表 1 *a* − *b* 部分检索结果对应的犹豫模糊集合

公式	P_S	P_N	P_O
$a-b$	{1,1,1,1}	{0.088,0.116}	{0.173}
$a-b=c$	{1,0.6,1,1}	{0.059,0.077}	{0.087}
$(a-b)^2$	{1,0.5,0.936,1}	{0.059,0.077}	{0.058}
$c(a-b)$	{1,0.5,0.876,1}	{0.059,0.077}	{0.058}
$\frac{a+b}{a-b}=\frac{c+d}{c-d}$	{0.230,0.2,0.768,0.7}	{0.044,0.058}	{0.049}

表 2　$a-b$ 部分结果表达式相似度计算结果

参数选择	$\lambda=1$	$\lambda=2$	$\lambda=5$
$a-b=c$	0.926	0.873	0.757
$(a-b)^2$	0.903	0.839	0.696
$c(a-b)$	0.898	0.836	0.695
$\frac{a+b}{a-b}=\frac{c+d}{c-d}$	0.767	0.653	0.451

排序结果为:

$sim_{\text{formulae}}(a-b,a-b=c)>sim_{\text{formulae}}(a-b,(a-b)^2)>sim_{\text{formulae}}(a-b,c(a-b))>sim_{\text{formulae}}(a-b,\frac{a+b}{a-b}=\frac{c+d}{c-d})$

由表 2 可以看出,表达式 $a-b$ 与表达式 $a-b=c$ 的相似度最大,与表达式 $\frac{a+b}{a-b}=\frac{c+d}{c-d}$ 的相似度最小,原因是子式 $a-b$ 在 $a-b=c$ 中的层次更低,水平位置更靠前。选取不同的 λ 值,计算所得相似度结果存在差异,但是并没有对排名结果产生明显的影响。

4　数学表达式相关排序方法对比

采用 C#编程语言,以 138 539 条从网络上获取的数学表达式为实验数据集,结合数学表达式检索系统,利用基于犹豫模糊集的数学表达式相似度评价方法,实现了检索系统的数学表达式检索结果排序模块。

由于目前可见的、明确具有相关排序功能的数学检索系统并不多见,为检验效果,将本文算法与同样具有包含匹配模式的由 Springer 设计研究的 LaTeX Search[32] 数学检索系统进行比较。选取 10 个不同的查询表达式,如表 3 所示,利用 LaTeX Search 对每个表达式进行检索,并对每一个表达式的结果集合运用本文算法进行排序,其中对表达式 $\sqrt{b^2-4ac}$ 的排序结果如表 4、表 5 所示。

表 3　查询表达式

序号	查询表达式	序号	查询表达式
1	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$	6	$\lim\limits_{n\to\infty}\frac{1}{n}$
2	a^2+b^2	7	$\frac{-b}{2a}$
3	$\sqrt{b^2-4ac}$	8	e^x-e^{-x}
4	$x\cos x$	9	$x+a$
5	$\sin^2 x$	10	$f\times g$

表 4　排序结果对比 1

序号	LaTeX Search	本文算法
1	$p_x^2=\frac{2c}{-b\pm\sqrt{b^2-4ac}}$	$\pm\sqrt{b^2-4ac}$
2	$\pm\sqrt{b^2-4ac}$	$x=(-b\pm\sqrt{b^2-4ac})/2a$
3	$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	$x=(-b\pm\sqrt{b^2-4ac})/2a$
4	$f_{\text{DB}}=\frac{(V_{\text{DB}}+V_{\text{SP}})}{V_{\text{DB}}A_T}\times\frac{-b+\sqrt{b^2-4ac}}{2a}$	$\frac{-b\pm\sqrt{b^2-4ac}}{2a}$
5	$C_{P\text{Baseline}}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	$AWCD=[\sum(C-R)]/[\sum(C-R)]n\times\sqrt{b^2-4ac}-n$
6	$\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	$\eta=\sqrt{\frac{2N-M}{2M-2N}}$, where $N=M(-b+\sqrt{b^2-4ac})/(2a)$ $a=-M_P,b=1-\frac{H_r}{H_0}-M_S-M_P,c=M_S+1-\frac{H_r}{H_0}$
7	$\eta=\sqrt{\frac{2N-M}{2M-2N}}$, where $N=M(-b+\sqrt{b^2-4ac})/(2a)$ $a=-M_P,b=1-\frac{H_r}{H_0}-M_S-M_P,c=M_S+1-\frac{H_r}{H_0}$	$X=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$
8	$w_{\text{cut}}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$

表 5 排序结果对比 2

序号	LaTeX Search	本文方法
9	$A_c = \frac{-b+\sqrt{b^2-4ac}}{2a}$	$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
10	$x = \frac{-b+\sqrt{b^2-4ac}}{2a}$	$x = \frac{-b+\sqrt{b^2-4ac}}{2a}$
11	$\int \frac{\mathrm{d}x}{\sqrt{a+bx+cx^2}} = \begin{cases} \frac{1}{\sqrt{c}}\operatorname{arcsinh}\left(\frac{2cx+b}{\sqrt{4ac-b^2}}\right), & \text{if } c>0 \text{ and } 4ac>b^2 \\ \frac{-1}{\sqrt{-c}}\arcsin\left(\frac{2cx+b}{\sqrt{b^2-4ac}}\right), & \text{if } c<0 \text{ and } 4ac<b^2 \end{cases}$	$A_c = \frac{-b+\sqrt{b^2-4ac}}{2a}$
12	$AWCD = [\sum(C-R)]/[\sum(C-R)]n\sqrt{b^2-4ac}-n$	$p_x^2 = \frac{2c}{-b \pm \sqrt{b^2-4ac}}$
13	$x_u = \frac{-b-\sqrt{b^2-4ac}}{2a}$ $r_u = x_u(\theta-\theta_0)+\theta_0 f_1$ $x_l = \frac{-c_1 b_2 + b_1 c_2}{a_1 b_2 - a_2 b_1}$ $r_l = \frac{-c_1 a_2 + c_2 a_1}{b_1 a_2 - b_2 a_1}$	$w_{\text{cut}} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
14	$x = (-b \pm \sqrt{b^2-4ac})/2a$	$x_u = \frac{-b-\sqrt{b^2-4ac}}{2a}$ $r_u = x_u(\theta-\theta_0)+\theta_0 f_1$ $x_l = \frac{-c_1 b_2 + b_1 c_2}{a_1 b_2 - a_2 b_1}$ $r_l = \frac{-c_1 a_2 + c_2 a_1}{b_1 a_2 - b_2 a_1}$
15	$x = (-b \pm \sqrt{b^2-4ac})/2a$	$C_{p\text{Baseline}} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
16	$z_l = \frac{-b-\sqrt{b^2-4ac}}{2a}$ $y_l = \frac{2z_l(n_z x_1 - n_x z_1) + n_x(x_1^2+y_1^2+z_1^2+r^2-\rho'^2)}{2(n_x y_1 - n_y x_1)}$	$z_l = \frac{-b-\sqrt{b^2-4ac}}{2a}$ $y_l = \frac{2z_l(n_z x_1 - n_x z_1) + n_x(x_1^2+y_1^2+z_1^2+r^2-\rho'^2)}{2(n_x y_1 - n_y x_1)}$
17	$x = -\frac{b}{2a} \pm \sqrt{\frac{b^2-4ac}{4a^2}} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	$P = -b \pm \frac{\sqrt{b^2-4ac}}{2a} = 315 \pm \frac{\sqrt{99\,225+1\,466\,080}}{1\,870}$ $= -315 \pm \sqrt{\frac{1\,565\,305}{1\,870}} = -315 \pm \frac{1\,251.121\,5}{1\,870}$ $= \frac{-1\,566.121\,5}{1\,870} = -0.837\,498$ 1*pt* after changing sign, $P = 0.837\,498$
18	$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	$x = -\frac{b}{2a} \pm \sqrt{\frac{b^2-4ac}{4a^2}} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
19	$X = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	$f_{\text{DB}} = \frac{(V_{\text{DB}}+V_{\text{SP}})}{V_{\text{DB}}A_T} \times \frac{-b+\sqrt{b^2-4ac}}{2a}$
20	$P = -b \pm \frac{\sqrt{b^2-4ac}}{2a} = 315 \pm \frac{\sqrt{99\,225+1\,466\,080}}{1\,870}$ $= -315 \pm \sqrt{\frac{1\,565\,305}{1870}} = -315 \pm \frac{1\,251.121\,5}{1\,870}$ $= \frac{-1\,566.121\,5}{1\,870} = -0.837\,498$ 1*pt*after changing sign, $P = 0.837\,498$	$\int \frac{\mathrm{d}x}{\sqrt{a+bx+cx^2}}$ $=$ $\begin{cases} \frac{1}{\sqrt{c}}\operatorname{arcsinh}\left(\frac{2cx+b}{\sqrt{4ac-b^2}}\right), & \text{if } c>0 \text{ and } 4ac>b^2 \\ \frac{-1}{\sqrt{-c}}\arcsin\left(\frac{2cx+b}{\sqrt{b^2-4ac}}\right), & \text{if } c<0 \text{ and } 4ac<b^2 \end{cases}$

为比较排序效果,由一组专家对每个查询表达式的检索结果集合进行人工排序作为评价标准,利用肯德尔相关系数分别计算本文算法和 LaTeX Search 算法的排序结果与专家排序结果的相关性,肯德尔相关系数越高说明该方法的排序结果与专家排序结果越接近,即更满足用户需求,以比较两者的排序效果,如图 5 所示。

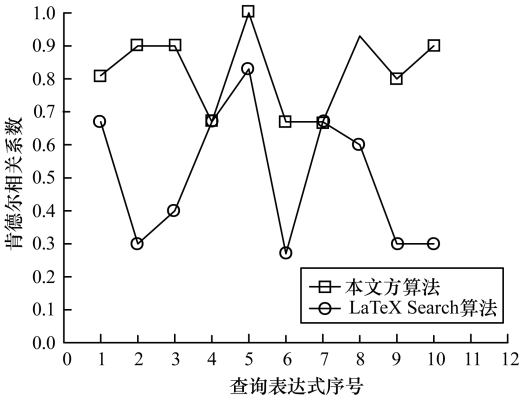

图 5　2 种算法与人工排序结果的相关性比较

由表 4、表 5 可以看出,本文系统将表达式 $\pm\sqrt{b^2-4ac}$ 排到首位,原因是该表达式中子式 $\sqrt{b^2-4ac}$ 的层次为 0,并且水平位置为 2(设起始位置为 1),显然表达式 $\pm\sqrt{b^2-4ac}$ 与其他表达式相比更贴近查询表达式。由图 5可以看出,在对选取的 10 个查询表达式结果集合进行排序的效果上,本文算法排序结果与人工排序结果的肯德尔相关系数更高,即本文算法更符合查询用户的需求。

为验证本文算法的时间效率,分别选取长度为 5,10,15,20,25,80 的数学表达式各 10 个,测试其在计算相似度过程中所需要的运算时间并取平均值,实验结果如图 6 所示。

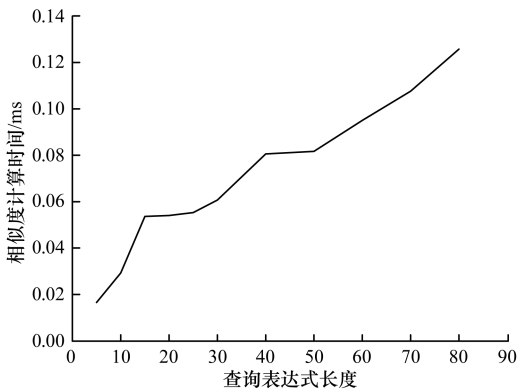

图 6　计算时间统计

由图 6 可知,当查询表达式长度为 5 时,平均计算时间为 0.016 4 ms;当查询表达式长度达到 80 时,平均计算时间为 0.125 8 ms,预计表达式长度达到一定长度时,其平均计算时间仍在可接受范围之内,不会为数学表达式查询系统增加过多的额外负担。

5　结束语

本文针对数学表达式的特点,提出一种基于犹豫模糊集的数学表达式相似度评价算法,设计数学表达式的评价属性、评价指标与相应的犹豫模糊隶属度函数,实现对数学表达式从结构、语义等多方面进行相似度评价,并将此算法应用于数学表达式检索系统检索结果的相关排序上。实验结果表明,所提出的算法能够实现数学表达式相似度评价以及对数学表达式检索结果的有效排序。下一步将尝试增加与完善数学表达式相似度的评价属性与评价指标,并对犹豫模糊隶属度函数中的相关参数选择进行调整和优化,不断改进相似度评价的效果。

参考文献

[1] Youssef A. Methods of Relevance Ranking and Hit-content Generation in Math Search[C]//Proceedings of the 6th International Conference on Towards Mechanized Mathematical Assistants. Berlin, Germany: Springer, 2007: 393-406.

[2] Shatanwi M, Youssef A. Equivalence Detection Using Parse-tree Normalization for Math Search [C]// Proceedings of the 2nd International Conference on Digital Information Management. Washington D. C., USA: IEEE Press, 2007: 643-648.

[3] Miner R, Munavalli R. An Approach to Mathematical Search Through Query Formulation and Data Normalization[C]//Proceedings of the 6th International Conference on Towards Mechanized Mathematical Assistants. Berlin, Germany: Springer, 2007: 342-355.

[4] MathWebSearch: Searching Math on the Web[EB/OL]. [2016-02-04]. http://search. mathweb. org.

[5] Kohlhase M, Sucan I. A Search Engine for Mathematical Formulae[C]//Proceedings of International Conference on Artificial Intelligence and Symbolic Computation. Berlin, Germany: Springer, 2006: 241-253.

[6] Kohlhase M, Sucan I, Jucovschi C, et al. MathWebSearch 0.4, a Semantic Search Engine for Mathematics[EB/OL]. (2010-12-22). http://mathweb. org/projects/mws/pubs/mkm08. pdf.

[7] Melis E, Haywood J, Smith T J. Leactivemath [C]// Proceedings of the 1st European Conference on Technology Enhanced Learning. Berlin, Germany: Springer, 2006: 660-666.

[8] Libbrecht P, Melis E. SemanticSearch in Leactivemath [EB/OL]. (2016-02-04). http://www. hoplahup. net/copy _ left/Libbrecht-etal-Semantic-Search-WebALT-06. pdf.

[9] Libbrecht P, Melis E. Methods to Access and Retrieve Mathematical Content in Active Math[C]//Proceedings of International Congress on Mathematical Software-ICMS. Berlin, Germany: Springer, 2006: 331-342.

[10] Egomath[EB/OL]. (2016-02-04). http://egomath. projekty. ms. mff. cuni. cz.

[11] Misutka J, Galambos L. Mathematical Extension of Full Text Search Engine Indexer[C]//Proceedings of the 3rd International Conference on Information and Communication Technologies. Washington D. C., USA: IEEE Press, 2008: 1-6.
[12] Mišutka J, Galamboš L. Extending Full Text Search Engine for Mathematical Content[C]//Proceedings of the 3rd Workshop on Digital Mathematics Libraries. Berlin, Germany: Springer, 2008: 55-67.
[13] Mišutka J, Galamboš L. System Description: EgoMath2 as a Tool for Mathematical Searching on Wikipedia. org[C]//Proceedings of the 10th International Conference on Intelligent Computer Mathematics. Berlin, Germany: Springer, 2011: 307-309.
[14] 景 珂. 网络数学搜索中的数学查询语言与索引的研究[D]. 兰州: 兰州大学, 2009.
[15] Guo Wei, Su Wei, Li Lian, et al. MQL: A Mathematical Formula Query Language for Mathematical Search[C]//Proceedings of IEEE Conference on Computational Science and Engineering. Washington D. C., USA: IEEE Press, 2011: 245-250.
[16] 刘志伟. 数学搜索引擎研究[D]. 兰州: 兰州大学, 2011.
[17] Hu Xuan, Gao Liangcai, Lin Xiaoyan, et al. WikiMirs: A Mathematical Information Retrieval System for Wikipedia[C]//Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries. New York, USA: ACM Press, 2013: 11-20.
[18] Lin Xiaoyan, Gao Liangcai, Hu Xuan, et al. A Mathematics Retrieval System for Formulae in Layout Presentations[C]//Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. New York, USA: ACM Press, 2014: 697-706.
[19] Sojka P, Liska M. Indexing and Searching Mathematics in Digital Libraries[C]//Proceedings of the 10th International Conference on Intelligent Computer Mathematics. Berlin, Germany: Springer, 2011: 228-243.
[20] 赵丹群. 现代信息检索[M]. 北京: 北京大学出版社, 2008.
[21] Torra V. Hesitant Fuzzy Sets[J]. International Journal of Intelligent Systems, 2010, 25(6): 529-539.
[22] Kamali S, Tompa F. Structural Similarity Search for Mathematics Retrieval[C]//Proceedings of International Conference on Intelligent Computer Mathematics. New York, USA: ACM Press, 2013: 246-262.
[23] Zanibbi R, Yuan B. Keyword and Image-based Retrieval of Mathematical Expressions[C]//Proceedings of Conference on Document Recognition and Retrieval XVIII. Berlin, Germany: Springer, 2011: 993-1004.
[24] Schellenberg T, Yuan B, Zanibbi R. Layout-based Substitution Tree Indexing and Retrieval for Mathematical Expressions[C]//Proceedings of Conference on Document Recognition and Retrieval XIX. Berlin, Germany: Springer, 2012: 263-271.
[25] Zhang Qun, Youssef A. An Approach to Math-similarity Search[M]. Berlin, Germany: Springer, 2014.
[26] 秦玉平, 唐亚伟, 伦淑娴, 等. 一种基于二叉树的数学公式匹配算法[J]. 计算机科学, 2013, 40(5): 251-252, 278.
[27] Xu Zeshui, Xia Meimei. Distant and Similarity Measures for Hesitant Fuzzy Sets[J]. Information Sciences, 2011, 181(11): 2128-2138.
[28] Xu Zeshui, Xia Meimei. Hesitant Fuzzy Entropy and Cross-entropy and Their Use in Multi-attribute Decision-making[J]. International Journal of Intelligent Systems, 2012, 27(9): 799-822.
[29] 陈树伟, 蔡丽娜. 区间值犹豫模糊集[J]. 模糊系统与数学, 2013, 27(6): 38-44.
[30] 蔡丽娜. 区间值犹豫模糊集及其在决策中的应用研究[D]. 郑州: 郑州大学, 2013.
[31] Tian Xuedong, Yang Songqiang, Li Xinfu. An Indexing Method of Mathematical Expression Retrieval[C]//Proceedings of the 3rd International Conference on Computer Science and Network Technology. Washington D. C., USA: IEEE Press, 2013: 574-578.
[32] LaTeX Search[EB/OL]. (2016-04-11). http://www. latex search. com.

编辑 顾逸斐

(上接第 203 页)

[11] Yong A. A Better FFT Bit-reversal Algorithm Without Tables[J]. IEEE Transactions on Signal Processing, 1991, 9(10): 2365-2367.
[12] James S W. A New Bit Reversal Algorithm[J]. IEEE Transactions on Acoustics Speech & Signal Processing, 1990, 38(8): 1427-1483.
[13] 贾 渊, 王俊波, 姬长英. FFT 快速整序算法的对比、改进及实现[J]. 电子科技大学学报, 2009, 38(2): 292-295.
[14] Evans D. A Second Improved Digit-reversal Permutation Algorithm for Fast Fourier Transforms[J]. IEEE Transactions on Acoustics Speech & Signal Processing, 1989, 37(8): 1288-1291.
[15] Evans D. A Second Improved Digit-reversal Permutation Algorithm for Fast Fourier Transforms[J]. IEEE Transactions on Acoustics Speech & Signal Processing, 1989, 37(8): 1288-1291.
[16] 汪海兵, 徐淑正, 杨华中. 基于查找表的单基 FFT 原址倒序算法[J]. 清华大学学报(自然科学版), 2008, 48(1): 43-50.

编辑 顾逸斐