

印刷体数学公式结构分析方法的研究

田学东, 李娜, 徐丽娟

(河北大学数学与计算机学院, 保定 071002)

摘 要: 印刷体数学公式识别是 OCR 技术的重要组成部分, 也是识别技术发展的瓶颈所在。在介绍公式识别技术发展现状的基础上, 针对结构分析这一公式识别的关键环节, 提出了一种基于基准线和字符间空白域特征的公式二维结构分析方法, 并将语义和语境分析策略融入其中。实验表明, 这种方法对公式结构分析具有较好的鲁棒性和应用前景。

关键词: 数学公式识别; 结构分析; 基准线; 空白域

Research on Structural Analysis of Mathematical Expressions in Printed Documents

TIAN Xuedong, LI Na, XU Lijuan

(College of Mathematics and Computer, Hebei University, Baoding 071002)

【Abstract】 Mathematical expressions recognition is an important part of OCR technology. It is also a bottleneck in the development of recognition technology. To the structural analysis stage, which is a crucial course in printed formula recognition, this paper proposes a method which makes use of baseline and operator range with syntax analysis based on the introduction of the development state of mathematical expressions recognition. In experiments, this method shows robust adaptability for the structure of mathematical expressions, and will have a good foreground.

【Key words】 Mathematical expressions recognition; Structural analysis; Baseline; Operator rang

1 概述

随着计算机应用和网络技术的发展, 印刷体公式识别成为当今信息化社会的研究热点。目前主流光学字符识别(Optical Characters Recognition, OCR)系统虽然对科技文献中的文字部分有较高的识别率, 但对于掺杂了字符、图表、公式的混合文档, 由于无法处理其中的图表和公式, 而使其识别效率明显降低。科技文献中含有大量的数学公式, 其所包含字符的复杂多样性及其灵活多变的字体和大小, 决定了数学公式不可能有规范的排版规则, 所以至今还没有一种 OCR 系统能够成功地解决这类问题。因此, 研究公式识别、分析与重构, 对于拓宽 OCR 系统的应用领域, 具有重要意义。

数学公式识别系统一般包括 4 个部分: 公式提取, 公式预处理与识别, 结构分析, 公式重构。其中结构分析是公式识别系统的关键环节。结构分析阶段产生的结构语法树是重构的主要依据, 直接影响系统的最终输出结果。另外, 对于结构分析阶段无法正确处理的字符可以反馈回公式识别阶段重新识别, 进一步提高系统的识别率。但是由于数学表达式的二维嵌套模型及其复杂的语义信息, 使结构分析成为公式识别系统的难点。

早在 20 世纪 60 年代就有人提出了公式识别问题。进入 90 年代, 这一领域的研究热度逐渐增加, 涌现出了许多关于数学公式的分析方法。这些方法大体可分为基于文法分析的方法^[1]和基于结构分析的方法^[2]。Anderson^[3]采用一种基于句法的分析方法, 通过使用并列语法来识别数学公式。Okamoto 和 Miyazawa^[4]使用递归地投影轮廓切分方法分析数学公式的结构。Grbavec 和 Blostein^[5]提出了用图重构的思想把公式中的结构信息用图的形式表示出来, 处理过程中根据构图规

则不断更新图, 从而完成整个公式的结构分析。

上述方法由于只针对某种类型的公式进行结构分析, 因此存在局限性, 不适用于所有的数学公式。本文提出了一种利用基准线特征分析公式嵌套结构, 并结合字符间空白域和语义环境对特殊字符和函数进行处理的方法。实验表明, 这种方法对数学公式的结构分析有较好的效果。

2 结构分析的预处理阶段

在数学公式中会出现一些彼此邻近并且关系密切的字符串, 它们用来表示一个独立的数学符号。为了便于结构分析, 需要先把它们提取出来在随后的分析中作为一个整体统一分析。而要完成这些功能, 须对字符的大小和中心进行归一化处理。

2.1 字符归一化

字符中心归一化首先需要计算字符的外边框, 并找出中心, 然后把字符中心移动到指定的位置上; 字符大小归一化对不同大小的字符做变换, 使之成为同一尺寸大小的字符。它是根据水平和垂直两个方向字符黑像素的分布进行的。先计算字符的质心 G_I 和 G_J :

$$G_I = \frac{\sum_{i=A}^B \sum_{j=L}^R i \cdot c(i, j)}{\sum_{i=A}^B \sum_{j=L}^R c(i, j)} \quad (1)$$

$$G_J = \frac{\sum_{i=A}^B \sum_{j=L}^R j \cdot c(i, j)}{\sum_{i=A}^B \sum_{j=L}^R c(i, j)} \quad (2)$$

基金项目: 河北省自然科学基金资助项目(F2004000132)

作者简介: 田学东(1963 -), 男, 教授, 主研方向: 中文信息处理与模式识别; 李娜、徐丽娟, 硕士生

收稿日期: 2006-03-30 **E-mail:** ruijia1982@163.com

式中 $c(i, j)$ 为 1 时表示该像素点为黑点(字符像素), 为 0 时表示该像素点为背景。再计算水平和垂直方向的散度 σ_i 和 σ_j :

$$\sigma_i = \sum_{i=A}^B \left(\sum_{j=L}^R c(i, j) \right) \cdot (i - G_i)^2 \bigg/ \sum_{i=A}^B \sum_{j=L}^R c(i, j) \quad (3)$$

$$\sigma_j = \sum_{j=L}^R \left(\sum_{i=A}^B c(i, j) \right) \cdot (j - G_j)^2 \bigg/ \sum_{i=A}^B \sum_{j=L}^R c(i, j) \quad (4)$$

最后, 按照比例将字符线性放大或缩小成规定散度的点阵^[6]。

2.2 字符串的提取

在数学公式中可以被看作一个字符独立处理的字符串称为字符串组。这样的字符串组通常包括两种: 复合型字符串组和结构型字符串组。复合型字符串组指处于同一水平线上的邻近字符集合, 它们表示一个独立的数学符号(比如十进制数字, 函数名等); 结构型字符串组指处于不同水平线上的通过彼此结构关系表示一个独立的数学符号的字符集合(比如分号、积分号等)。

(1) 复合型字符串的提取

用上述方法对字符归一化后, 运用最长字符串匹配的方法, 提取复合型字符串, 并把它看作一个整体符号。根据统计常用的函数名有 32 种: Arth, Arch, Arcth, Arsech, Arcsch, Arcsh, arcsin, arcsec, arctan, arcos, arccsc, arcctg, cos, ctg, csc, csch, ch, cth, exp, grad, log, ln, lg, lim, tan, th, sin, sec, sh, sech, rot, div。提取出的复合型字符串应为上述函数名集合的元素。

(2) 结构型字符串的提取

为了确定特殊符号的范围, 需要利用上述方法对字符归一化, 并利用坐标信息提取结构型字符串。例如根据分号的宽度信息可以把分号上下区域的字符提取出来作为一个独立的字符处理。

3 结构分析

结构分析阶段的输入是带有位置信息的字符序列, 输出是一棵语法树。整个过程分为 3 个阶段:

(1) 寻找公式的主基准线;

(2) 以主基准线为标准, 递归地寻找其它基准线, 得到字符间的嵌套关系;

(3) 将分析得到的信息用一棵语法结构树表示出来。

其中语法树中的每一个节点表示一个字符, 除了字符自身的属性信息外, 还包括 6 个位置指针, 分别代表 6 种不同的空间关系, up、super、right、subsc、down、inclusion 分别表示上部、上标、水平、下标、下部以及包含关系。其结构如图 1 所示。

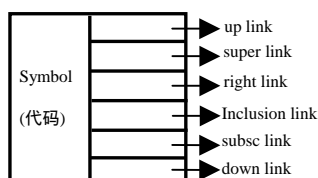


图 1 语法树节点结构

3.1 基准线

一个数学公式的基准线是指数学符号沿水平方向排列所在的直线区域。例如公式 $x^2 + 3y^{a+b} = c$ 中包含两条基准线。其中“x”, “+”, “3”, “y”, “=”, “c”在一条基准线₁上; “2”, “a”, “+”, “b”在另一条基准线₂上。

公式中基准线分为主基准线和嵌套基准线。主基准线上包含的字符与公式中其他字符没有任何被嵌套的关系。一般情况下公式中最左边的字符所在的基准线即为主基准线, 该字符称为首字符。例如上例中基准线₁为主基准线

嵌套基准线上的字符, 或在垂直方向上偏离了某个字符, 或被某个字符所包围。嵌套基准线常用来表示操作数的作用域范围。例如在公式 $\frac{a+b}{c+d}$ 中“a”, “+”, “b”位于同一个嵌套基准线, “c”, “+”, “d”位于另一嵌套基准线, 并且它们都附属于分号, 也就是分号的作用域包括“a”, “+”, “b”, “c”, “+”, “d”。

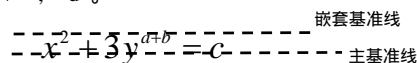


图 2 嵌套基准线

3.2 结构分析

结构分析概括为以下几步:

Step1 对输入的字符序列进行排序。

初始输入的字符序列中每个字符带有 6 个空间坐标信息 (min_x, max_x, center_x, min_y, max_y, center_y)。首先把初始字符信息存储在数组变量 myArray 中, 并用冒泡排序方法对数组进行排序, 使各字符按照 min_x 数值从小到大排列。这样便于在今后的分析中按照阅读习惯从左到右遍历整个公式。

Step2 确定公式中的主基准线。

根据定义, 公式中第 1 个字符所在的基准线即为主基准线。由于排版的不同, 一个基准线上字符的 center_y 不可能完全相同, 可以设定一个变化范围来描述字符中心所在的水平线区域。设阈值范围为 (center_y - tH, center_y + tH), 其中, H 是首字符的高度, t 是阈值 (0 < t < 0.5)。把中心点在这个阈值范围内的字符都认为是同一条基准线上。

设计了 BuildBaseLine() 函数寻找主基准线上的所有字符, 其算法描述如下:

(1) 按照图 1 的结构初始化一棵语法树 StructuralTree, 并定义一个指向这棵树的指针 Start;

(2) 把 myArray 中的第 1 个元素作为首字符, 即为标准字符;

(3) 遍历 myArray 中的每一个元素, 寻找 center_y 位于 (H/2 - tH, H/2 + tH) 范围内的其他字符, 这些字符所在的线即为主基准线;

(4) 把这些字符依次放入 StructuralTree 中, 彼此用 right 指针相连;

(5) 把这些字符从 myArray 中删除。

Step3 嵌套基准线的查找

嵌套基准线的查找是整个公式结构分析的关键部分, 也是难点所在。公式识别之所以难于文字识别就在于它的空间二维结构。通过嵌套基准线的查找就可以把字符间的空间结构关系转化为数学上符号间的逻辑关系。

递归地调用 NestBaseLine() 函数寻找整个公式的嵌套基准线。其步骤如下:

(1) 以当前 myArray 数组中的第 1 个元素为目标字符, 寻找其所在的嵌套基准线;

(2) 遍历 myArray 中的每一个元素, 寻找 center_y 位于 (H/2 - tH, H/2 + tH) 范围内的其他字符, 其中的 H 为目标字符的高度, 并将其存储在一个缓冲数组 bArray 中;

