法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-07-02

文書内における数式の構造認識

OGAWA, Koichi / 竹内, 朋之 / TAKEUCHI, Tomoyuki / 尾川, 浩一

(出版者 / Publisher)
法政大学計算科学研究センター
(雑誌名 / Journal or Publication Title)
Bulletin of Computational Science Research Center, Hosei University / 法政大学計算科学研究センター研究報告
(巻 / Volume)
17
(開始ページ / Start Page)
85
(終了ページ / End Page)
89
(発行年 / Year)
2004-03-22
(URL)
https://doi.org/10.15002/00025012

文書内における数式の構造認識

竹内　朋之　　尾川　浩一
法政大学大学院工学研究科

数式を含む文書を認識しコード化することは、文字や画像のみによって構成される文書を処理することよりも困難である。これは、数式を構成するシンボル群が特定の規則に基づいて配置されているためであり、この認識においてはシンボル等の空間的位置関係を用いてその構造を把握しなければならない。本研究では、数式におけるシンボルや式の空間的位置関係から構造を認識し、TeX コードを出力するシステムを開発した。

1. はじめに

科学技術文書をスキャナで走査し、得られた濃淡画像の中から数式が記述されている部分を抽出し、それを認識してすべてコード化できれば、印字品質を落とさないでプリンタへの出力が可能になったり、web 等への表示の取り扱いが簡単になる。ところが、一般の OCR ソフトでは数式部分は画像として取り扱われてしまい、認識の対象から外されてしまうのが実状である。このような数式を認識するシステムは、いくつかのグループによって研究が進められてきた[1]-[3]。一般に、数式認識は次のステップから成る。

(a) 文書中の数式領域の抽出
(b) 前処理(雑音除去や歪み補正等)
(c) シンボルの孤立化、区分け
(d) シンボルの認識
(e) シンボル間の空間的関係の同定
(f) シンボル間の論理的関係の同定
(g) 意味解釈

本研究では入力として、数式画像の領域を抽出した後の画像を用い、前処理(b)を施した後、シンボルの分離(c)と認識(d)を行い、構造を認識(e)-(g)した後に TeX コードを出力するシステムを作成することを目的としている。

2. 数式の構造

2.1 数式の記述規則

本研究では数式を以下のようにモデル化して取り扱う。

(1) 数式は､原則として左から右へと並ぶ文字(記号)の列である。これを文字(記号)の通常の接続とし、水平接続と呼ぶことにする。
(2) 数式は、途中で別の数式を含み、水平接続とは異なる方法で接続され得る。これによって派生した数式を付属式と呼ぶことにする。
(3) 付属式は派生元となるただひとつの親文字もしくは親付属式を持つ。
(4) 付属式と親文字との接続関係は様々あるが、本論文では内包、上方接続、下方接続、右上接続、左上接続の５つに分類する。すべての接続関係はこれらか水平接続としか見なさず、これ以外の接続関係は扱わない。
(5) これらの関係によって数式は木構造で表現される。

2.2 水平接続

例えば「5+3x」という式があるとき、「5」と「+」、「+」と「3」、「3」と「x」はそれぞれ水平接続の関係になる。

2.3 付属式

本論文では矩形の位置関係のみから付属式を検出する。よって

$$\overline{a} \quad \frac{x+1}{z} \quad \sum_{k=1}^{n}$$

などの数式はいずれも「縦に繋がった式」という属性のみが考慮され、分数とか総和記号といった意味の違いについては考慮しない。付属式は以下に示す内包、上方接続、下方接続、右上接続、右下接続の５種類のいずれかで接続される。

(1) 内包
親矩形の内側に存在する式で平方根がこれに当たる。

$$\sqrt{x}$$

(2) 上方接続
親矩形の上側に式が存在するもので、アクセント記号、分数の分子、総和記号の上限式などがこれに当たる。

$$\overline{a} \quad \frac{x}{a} \quad \sum^{x}$$

(3) 下方接続
親矩形の下側に式が存在するもので、分数の分母、総和記号式の下限式などがこれに当たる。

$$\frac{a}{x} \quad \sum_{x}$$

(4) 右上接続
親矩形の右側上部に式が存在するもので、累乗の指数、積分式の上限がこれに当たる。

$$e^{x} \quad \int^{x}$$

(5) 右下接続
親矩形の右側下部に式が存在するもので、下付き文字、積分式の下限がこれに当たる。

$$e_x \quad \int_x$$

この他にも左上接続や左下接続などが考えられるが、頻度が低く、アルゴリズムの簡単化のためここでは扱わないことにする。

3. 処理の流れ

本研究では以下の手順で数式画像の構造認識を行い、TeX コードの出力を行う。

(1) 数式画像の入力
(2) シンボルの分離
  (a) 特定の閾値で二値化
  (b) 画素の連結成分の全方向探索
  (c) i、j、!などにおける・の統合
  (d) 雑音の除去（孤立性の点など）
  (e) イコール記号の統合
(3) パターンマッチングによるシンボル認識
(4) 構造認識
  水平接続と付属式への接続を以下の手順で再帰的に行う。
  (a) 式の先頭の文字を探す。
  (b) 先頭から順に水平接続される文字を選択し、次々と繋げていく。この時点では付属式となるシンボルがあってもそれを無視する。
  (c) 水平接続の終点まで来たら、折り返して付属式となるシンボルを探す。
  (d) 途中で付属式となるシンボルを見つけたら、一旦、(a)に戻ってその式を処理する。式の終端に着いたらそこから、(c)に戻って引続き付属式を探す。
(5) 雑音の除去(接続の矛盾から判断する)
(6) 構造認識結果を示す画像を出力
(7) TeX コードを出力

3.1 画像入力および初期処理

数式画像は書籍をスキャナ（300dpi）で８bit にディジタル化したものを用い、数式部分を矩形状に切り出しておく。(Fig.1)

3.2 シンボルの分離

3.2.1 二値化と矩形抽出

画像が入力されたらまず適当な閾値(例えば、0 を黒、255 を白とする 255 階調で 192)で二値化し､その後に数式の最小単位である文字の矩形を得る。数式の認識はトップダウン方式で行うが、画素の集合を文字としてまとめる作業だけはここでやっておく。

$$\rho = \frac{\left| \int_{-\infty}^{\infty} C \frac{M^*(s)}{P_n(s)} M(s)\,ds \right|^2}{\int_{-\infty}^{\infty} C^2 \frac{M(s)^* M(s)}{P_n(s)^* P_n(s)} P_n(s)\,ds}$$

Fig.1 An original image

3.2.2 画素の連結成分の全方向探索

画素値によって全方向探索を行ない、連結成分単位で分ける。画質が十分に良ければこの時点でシンボルの孤立化は、ほぼ完了する。

3.2.3 分離文字の統合と雑音の除去

入力画像に「i」や「=」といった分離記号があった場合、それらは二つ以上の部分に分かれてしまっている。幸いにして数式に使われるシンボルはその形状がシンプルなものが多く、漢字やひらがなと違って分離記号は特定のものに限られる。これを捜し出して統合する。このとき画像中に含まれる雑音も取り除く。「i」や「j」や「!」といった分離記号は、その一部である点が雑音と間違われやすいので、雑音の除去よりも先に統合を行ない、その他のシンボルは後で統合する。(Fig.2)

$$\rho = \frac{\left| \int_{-\infty}^{\infty} C \frac{M^*(s)}{P_n(s)} M(s)\,ds \right|^2}{\int_{-\infty}^{\infty} C^2 \frac{M(s)^* M(s)}{P_n(s)^* P_n(s)} P_n(s)\,ds}$$

Fig.2 Rectangular regions including each symbol.

3.3 パターンマッチングによるシンボル認識

パターンマッチングを用いて個々のシンボルを認識した後、その結果に応じてシンボルの中心座標を求めて後の処理に役立てる。中心座標は基本的にはそのシンボルを囲む矩形の縦横の中心をそのまま使う。ただし「b」や「h」のように上に突き出た文字はそれより若干低めに、「g」や「y」のように下に突き出た文字は若干高めに中心座標を設定する。

3.4 構造認識

数式はある親シンボルに子として式が付属し、その式を構成する文字にもまた子の式が付属し得るといったような、木構造を持ったオブジェクトである。本アルゴリズムは数式のこの構造を再帰アルゴリズムによって走査・認識している。

3.4.1 シンボル同士の空間的関係

ここで、ある二つの位置も大きさも異なるシンボルの矩形 A、B が与えられたときの、それらの空間的関係を判断するためのアルゴリズムを示す。この時、親シンボル B に子シンボル A が付属するものとする。

I. 二つの矩形の重なる面積が A の面積の 80%以上ならば内包と判断

II. 二つの矩形が Y 軸への射影で全く重なっていなければ以下のいずれかと判断
   ① X 軸への射影で見て A が B に対し完全に（全く重ならずに）左だったら無関係
   ② X 軸への射影で A が B に完全に包含されていれば以下のいずれか
      - Y 軸への射影で A が B に対し完全に上だったら上方接続
      - Y 軸への射影で A が B に対し完全に下だったら下方接続
   ③ 二つの矩形が X 軸への射影で少しでも重なっていれば以下のいずれか
      - A の横幅が B の横幅×４／３より大きければ無関係
      - Y 軸への射影で A が B に対し完全に上だったら上方接続もしくは右上接続のいずれか
      - Y 軸への射影で A が B に対し完全に下だったら下方接続もしくは右下接続のいずれか
   ④ 二つの矩形が X 軸への射影で完全に右だったら以下のいずれか
      - Y 軸への射影で A が B に対し完全に上だったら右上接続
      - Y 軸への射影で A が B に対し完全に下だったら右下接続

III. 二つの矩形が Y 軸への射影で少しでも重なっていれば以下のいずれかと判断
   ① A の中心が B の中心より左だったら無関係
   ② A の上端が B の上端より上だったら以下のいずれか
      - A の下端が B の中心より上だったら右上接続
      - A の下端が B の下端より上で、かつ A の中心が B の上端と中心の中間より上ならば右上接続
   ③ A の下端が B の下端より下だったら以下のいずれか
      - A の上端が B の中心より下だったら右下接続
      - A の上端が B の上端より下で、かつ A の中心が B の下端と中心の中間より下ならば右下接続
   ④ ２つの矩形の中心の位置関係について、X 軸での距離が Y 軸での距離の 5 倍より大きければ水平接続
   ⑤ A の下端が B の中心より上だったら右上接続
   ⑥ A の下端が B の下端より上で、なおかつ A の中心が B の上端と中心の中間より上なら右上接続
   ⑦ A の上端が B の中心より下だったら右下接続
   ⑧ A の上端が B の上端より下で、なおかつ A の中心が B の下端と中心の中間より下なら右下接続

IV 上のいずれの条件にも当てはまらなければ水平接続

3.4.2 再帰的構造認識

シンボル同士の空間的関係の判断ができたら、それらのサイズ、位置情報等からそのシンボルの『基幹あるいは枝葉となる式における先頭文字の適合性』を評価する。左にあり、付属シンボルを持ちそうな矩形を多く持つシンボルほど、『適合性』が高いということになる。その評価値が最も高いものを、その式における基幹式の先頭のシンボルとする。

先頭のシンボルが決定したら、水平接続されるシンボルを探して右方向へと繋げていく。これでその式の基幹式となるシンボル列が決定する。途中で上下や斜め方向(あるいは平方根のように『内側』)にあるシンボルがあればそれを付属式として接続しなくてはならない。だが、その付属式となるシンボルが基幹式のどのシンボルに接続されるべきなのかが分からないという問題がある。例えば、

$$m\frac{1-n}{2-n}$$

という式があったとき、『1』は『m』の右上に接続される可能性がある。そういった場合には、基幹式において後ろ(右側)のシンボル（上の例の場合では分数線）を優先してその付属式の親矩形として選択することで、付属シンボルの配属先を確定する。基幹式のシンボル列のうち右のものから順にそれに付属するシンボルを探すようにすれば、この規則は自ずと守られる。付属シンボル群を見つけた際、それ自体を式とみなし、以上の処理を再帰的に行う。

木構造を作成するための手順をまとめて以下に示す。これによって、個々のオブジェクトの関係を調べ、一体となった構造を作る。

(a) 式の先頭の文字を探す。
(b) 先頭から順に水平接続される文字を選択し、次々と繋げていく。この時点では付属式となるシンボルがあってもそれを無視する。
(c) 水平接続の終点まで来たら、先頭に戻り付属式となるシンボルを探す。
(d) 途中で付属式となるシンボルを見つけたら、(b)-(c)の手順でその式を処理する。式の終端に着いたらそこから元の式に戻り、引続き付属式を探し処理を続ける。

3.5 雑音の除去

構造認識においてどこにも属さなかったシンボルは無視する。また接続関係において矛盾しているものは排除し、不確定のままになっている関係はここで確定する。

3.6 構造認識結果を示す画像を出力

シンボルの空間的関係をどのように決定したのかを視覚的に提供する。Fig.2 で示した例の認識結果を Fig.3 に示す。

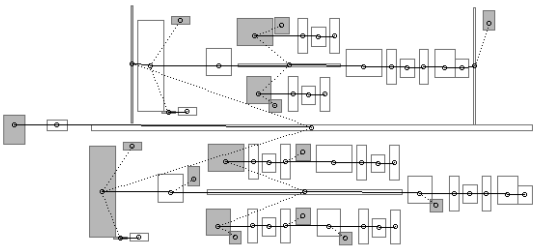

Fig.3　Structure of an mathematical expression

3.7 TeX コードの出力

構造認識とパターンマッチングの結果を元に、TeX のコードを出力する。下は出力された TeX コードであり、Fig.4 はそのコンパイル結果である。

```
¥[ ¥rho =¥frac{¥vert ¥int ^{¥infty }_{-¥infty }C¥frac{M^{¥ast }(s)}{P_{n}(s)}M(s)ds¥vert ^{2}}{¥int ^{¥infty }_{-¥infty }C^{2}¥frac{M(s)^{¥ast }M(s)}{P_{n}(s)^{¥ast }P_{n}(s)}P_{n}(s)ds} ¥]
```

$$\rho = \frac{\left| \int_{-\infty}^{\infty} C \frac{M^{*}(s)}{P_n(s)} M(s) ds \right|^2}{\int_{-\infty}^{\infty} C^2 \frac{M(s)^{*} M(s)}{P_n(s)^{*} P_n(s)} P_n(s) ds}$$

Fig.4　Mathematical expression reproduced by TeX.

4. 結果と考察

以下に、本プログラムの出力結果の例を示す。Fig. 5 は抽出された矩形とその構造である。これより、出力された TeX コードは以下のようになった。また、それよりコンパイルされた数式を Fig.6 に示す。

```
¥[ f(x,y)=¥frac{1}{2}¥int ^{2¥pi }_{0}¥frac{1}{U^{2}}¥int ^{¥infty }_{-¥infty }P_{f}(¥beta ,s)g(s'-s)¥frac{D}{¥sqrt{D^{2}+s^{2}}}dsd¥beta ¥]
```

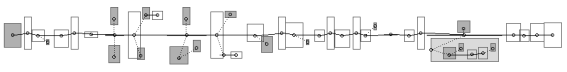

Fig.5 Example 1 (structure of the mathematical expression)

$$f(x,y) = \frac{1}{2} \int_0^{2\pi} \frac{1}{U^2} \int_{-\infty}^{\infty} P_f(\beta, s) g(s' - s) \frac{D}{\sqrt{D^2 + s^2}} ds d\beta$$

Fig.6　Example 1 (Reproduced expression)

Fig. 7 は第二の例の抽出された矩形とその構造である。これより、出力された TeX コードは以下のようになった。また、それよりコンパイルされた数式を Fig. 8 に示す。

```
¥[ ¥lambda ^{n+1}_{j}=¥frac{¥lambda ^{n}_{j}}{¥displaystyle{¥sum_{i¥in J_{j}}}c_{ij}+¥frac{1}{¥beta }¥frac{¥partial U(¥lambda ^{n})}{¥partial ¥lambda _{j}}}¥displaystyle{¥sum_{i¥in J_{j}}}¥frac{c_{ij}P_{i}}{R^{n}_{i}} ¥]
```

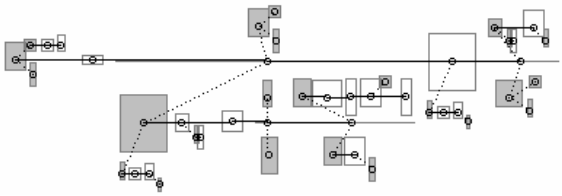

Fig.7. Example 2 (structure of the mathematical expression)

$$\lambda_j^{n+1} = \frac{\lambda_j^n}{\displaystyle\sum_{i \in J_j} c_{ij} + \frac{1}{\beta} \frac{\partial U(\lambda^n)}{\partial \lambda_j}} \sum_{i \in J_j} \frac{c_{ij} P_i}{R_i^n}$$

Fig.8. Example 2 (Reproduced expression)

使用した数式 21 個のうち、16 個は適切に認識したが、5 個は失敗した。失敗した 5 個のうち 1 個は入力画像の画質の不十分さからシンボルの分離に失敗したもので、残り 4 個はいずれも個々のシンボル認識の失敗だった。本研究の狙いは、数式構造の認識であり、シンボルの認識には単純なパターンマッチングによる手法を用いており、シンボル認識の精度は向上させることが可能であると考えている。これらに対し、空間的構造の認識はすべて成功した。しかし、入力画像の条件によっては失敗することもありうる。たとえば、シンボル同士が接触するばあいであり、入力時の空間分能が低い場合や雑音によってこのような問題が発生する。実用化を目指すためにはこのような問題を解決する必要がある。本研究で提案した方法は、非常に単純なアルゴリズムであり、このため実行速度が非常に速いというのがメリットとなっているが、精度をさらに高めるには、構造に関しての論理的解釈をすべて取り入れ、再帰的に確認する必要があると考える。

参考文献

[1]Dorothea Blostein, Ann Grbavec 、 "Recognition of mathematical notation", Handbook of Character Recognition and Document Image Analysis, pp. 557-582 Eds.H.Bunke and P.S.P.Wang、1997.

[2]鈴木昌和, 玉利文和, 井上浩一, 宮崎亮乃輔, 宮平彩乃、”OCR を用いた科学技術文書の自動点訳について”、信学技報 TECHNICAL REPORT OF IEICE. HCS97-8, pp.7-14、1997-09.

[3]岡本正行, 東裕之、”記号のレイアウトに注目した数式構造認識”、電子情報通信学会論文誌 D－Ⅱ Vol.J78－D－Ⅱ No.3 pp.474-482、1995 年 3 月

キーワード.

数式表現、シンボル、認識

Summary.

Recognition of Structure of Mathematical Expressions in a Document

Tomoyuki Takeuchi　Koichi Ogawa
Graduate School of Eng., Hosei Univ., Tokyo, Japan

This paper proposes a new method for recognizing a structure of mathematical expressions in a document. This recognition method is based on the location of symbols which appear in the mathematical expression. The newly developed algorithm was tested by means of samples extracted from a textbook and the accuracy of recognition was evaluated.

Keywords.

Mathematical expression, Symbols, Recognition