数理解析研究所講究録
1169 巻 2000 年 92-96

A Recognition Method of Matrices by Combination Matching of Variable Block Pattern Elements Generating Rectangle

—矩形領域を生成する可変ブロックパターン要素の

組み合わせマッチングによる行列認識手法—

九州大学大学院　数理学研究科
金堀 利洋

1　はじめに

ネットワークやコンピュータの加速的な進歩に伴い、様々な可能性が生まれてきている。これは数学の分野や教育においても例外ではなく、数式処理システムや学術情報データベースに接続した電子ボードなど、数学の授業や講義に新しいスタイルをもたらすと考えられている。このようにして、数式をコンピュータ上で取り扱うことの必要性が重要視され、研究が進んでいる。しかしながら、数学に欠かせない“行列”については未だそれを扱う環境は整っていないと言って良い。

1.1　コンピュータにおける行列の取り扱い

数式を扱う環境として主に、数式処理システム、TEX や MathML などの数式フォーマット、OCR などが挙げられる。それぞれについて、行列の取り扱いに関する問題点を挙げると、前者 2 つは、ユーザーが入力したり、処理または表現できる行列が非常に限られている。

次に OCR についてであるが、行列以外の数式に関しては研究が進んでいて、[1] では 2 次元構文解析を用いた数式認識を紹介している。しかし、これは計算量の面等からあまり実用的でない。実用的なものとしては、[2]、[3]、[4]、[5]、などは実用的なレベルでの数式認識を可能としている。しかしながら、これらは行列をまったく含まない数式か、含んでいても非常に限られた行列しか認識できない。例えば、要素が行、列にきちんと区切られている等、整形されたもののみを認識対象としており、数学などで普通に用いるような略記をした行列は全く認識できない。仮に、行列の認識が出来たとしても、これは前述の数式を取り扱う 2 つの環境とも関連することであるが、行列の構造を保持しておくこと、つまり、その行列を表現することが出来ないのである。

1.2　目的

上で述べたように、OCR において認識できる行列は非常に限られている。しかし、行列は数学においては欠かせないものである。よって、あらゆる行列を認識できる事が望ましく、その為に今回は新たな行列認識のアルゴリズムを提案する。

尚、今回提案するアルゴリズムは現段階 (2000 年 4 月) ではまだ実装されておらず、今後、早急に実装する予定である。この事を前もって了承されたい。

2 行列認識の定義

まず、行列認識の定義の前に、先ずはその前提と動機を述べる。

2.1 前提

行列の認識を行う際の前提条件を以下のように定義する。

前提 認識する行列において、その構成要素それぞれが適切に一つずつのブロックとして認識され、その位置と大きさ、外見的特徴が正しく認識されている状態。

これは下にある行列のように、一見離れていたり、複数列 (行) に跨っている様に見えても、1 つの構成要素なら、1 つのブロックとして扱われ、更に、位置と大きさが正しく取り出され、誤認識が全く無い状態という事である。

$$\begin{pmatrix} \frac{1}{J_r}(a_1+\cdots+a_n) & & 0 \\ & \ddots & \\ 0 & & \frac{1}{J_r}(a_1+\cdots+a_n) \end{pmatrix} \Longrightarrow \begin{pmatrix} \boxed{\frac{1}{J_r}(a_1+\cdots+a_n)} & & 0 \\ & \ddots & \\ 0 & & \boxed{\frac{1}{J_r}(a_1+\cdots+a_n)} \end{pmatrix}$$

$$\begin{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} & B \\ B^T & \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \end{pmatrix} \Longrightarrow \begin{pmatrix} \boxed{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}} & B \\ B^T & \boxed{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}} \end{pmatrix}$$

一見するとこの前提条件は現実的では無いと思われるかもしれない。しかし、全てを自動化する必要は必ずしも無く、この前提はユーザの介入によって成立するものとしてもよい。例えば、文書中の行列に対して、ユーザがマウス等によって構成要素のブロックを指定し、その指定された範囲それぞれを数式認識にかけ、誤認識・構文解析のミスなどがあれば、ユーザが修正を加える、というような操作で前提は成立する。

2.2 動機と定義

通常、行列を表記する時、その要素を一つ一つ全て書くのではなく、何らかの規則[1]に基づいた略記を行う。ここで、我々は "略記" を「元の行列の持つ矩形領域をいくつかの領域に分け、そのそれぞれの領域を代表元、もしくはその領域の形状等を表す記号で表記したもの」と考えた。つまり、略記された行列 (**略記行列**) とは、(1) 矩形領域の切り分け、(2) 切り分けた領域の代表元、記号への置き換え、により表されたものと定義する。すると逆に、「表記された行列の代表元、記号のとりうる領域から、矩形領域を生成」することが出来れば、元の行列の構成が理解できると我々は考えた。

行列の構成の理解が出来れば、何らかの文法による略記行列の一意的な記述や、更に踏み込んで、数式としての行列の解析、また、編集・検索などの再利用性の付加など、現在 Web 上などで単に画像として扱われていた行列に様々な可能性が与えられる。

ここで、上記の "行列の構成の理解" を行列認識の定義とする。

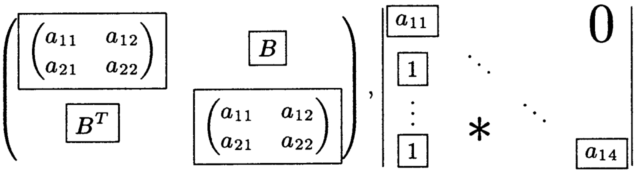
3 略記行列の構成要素

今回、我々が実際の印刷物から採集した行列 (約 150 超) から分類した、略記行列の構成要素の要素名とその機能・性質を挙げる。

[1] この規則を見出すことが我々の目的であると言ってよい。

数式要素

- 行列の要素を表す基本単位。
- 略記行列によって表される行列 (**表記行列**) の要素そのままであったり、規則性を表すものであったりする。

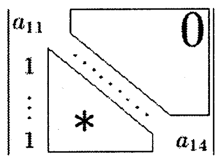


- 1 行 1 列の領域をとる。
- 行列を含む数式からなる。

ここで、**表記行列**とは、略記行列によって表される (表したい) 行列の事で、そのままでは具体的に記述することが出来ないものもある。例えば、行数や列数が定まっていない場合など。

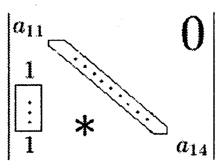
領域記号

- よく現れるのは、O、0、や、I、1。
- 現在のところ上記以外の領域記号として小文字が現れることはほとんどない。
- また、空白も同様に領域を持つものとして考える。



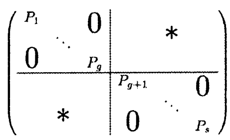
連続記号

- 略記行列内で、一定の方向にある規則性を持って要素が並んでいることを表す。
- 連続している個数が修飾記号を用いて明示的に指定されているものや、暗黙的に指定されている場合や、個数が指定されていないものがある。



仕切り記号

- 行列の矩形領域の分割の様子を表す。
- 行列内に配置される。



修飾記号

- 連続記号や、仕切り記号の個数や大きさを更に詳しく表記したい場合。
- 行や列を指定する場合。
- その他、他の記号に何らかの付加的な意味を持たせたい場合に用いられる。

修飾記号は今回は認識の対象外とする。その理由として、修飾記号は行列内に領域を持たないので、先に述べた行列構成の理解のアイデアには含まれない上に非常にバリエーションが豊富であるため、修飾記号までカバーしようとすると、アルゴリズムが複雑になってしまう。結果、基礎となるべきアルゴリズムを提案するという、今回の目的を外れてしまうからである。

次に、行列認識の際に必要だと思われる構成要素のパラメータを挙げる。

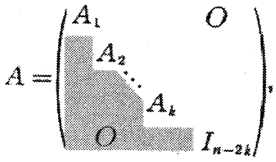
外見的特徴

- 実際に印刷されている形状や大きさ。
- 機能的特徴が同じでも、印刷物によって違ってくるものがある。

機能的特徴 その記号の略記行列における役割を表すものであるが、更に以下に分けられる。

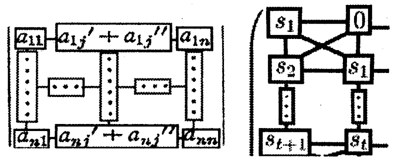
取り得る領域の形状

- その要素が代表する領域のパターン。
- 可変の大きさを持つ。
- 他の領域と交わることはない。



接続条件

- 他の要素との位置関係。
- 連続記号の両端は、数式要素か、または方向の異なる連続記号であり、稀に行列の端の場合もある。
- 数式要素は 8 方向に接続する。



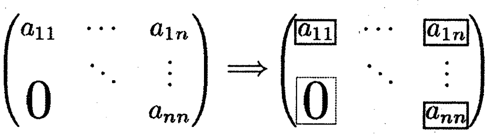
優先順位

領域を割り当てる際の順番

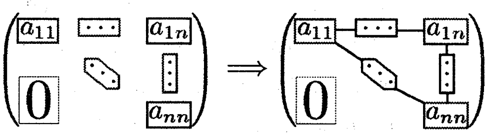
4　行列認識のアルゴリズム

ここで、我々の考えた行列認識のアルゴリズムを紹介する。

1. 数式要素を定める パラメータより数式と思われる構成要素を確定する。もちろん、代表元となりうるものは区別しておく。

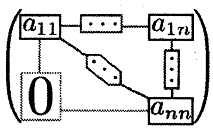


2. 連続記号を定める パラメータより連続記号と思われる構成要素を確定する。その際、接続条件から、他の確定した構成要素との接続状況、位置関係を定める。

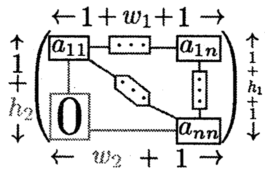


3. 連結成分間の位置を定める 数式要素の接続条件から接続関係を求める。その際、接続条件から他の確定した構成要素との接続状況、位置関係を求める。この時、既にある接続条件と矛盾し、更に、その対象が孤立した数式要素があれば、それを代表元とみなし接続する。

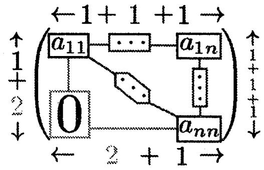
右図において 0 を数式要素として a_{11} と a_{nn} に接続をすると、既に確定しているパス "$a_{11} \to$ 連続記号 $\to a_{nn}$" と矛盾が生じる。



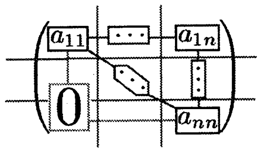
4. 外接矩形の大きさを求める 定まった接続関係から可変長を含んだ外接矩形の大きさを求める。仕切り記号がある場合はまず、仕切られた領域の大きさを求める。



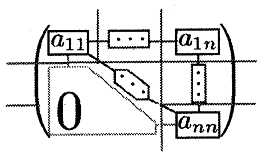
5. 可変長部分の長さを決定 4. で求めた外接矩形で最小のものを仮の矩形領域とし、可変長部分の長さを決定する。



6. 代表元以外に領域を割り当てる 仮の矩形領域上の領域を代表元以外の構成要素に割り当てる。仕切り記号がある場合はそれに従って領域を仕切る。



7. 残りの要素に領域を割り当てて終了 要素が配置されていない領域内に代表元があり、その領域がパターンとマッチすればその領域を割り当てる。マッチしない場合は 2. からやり直す。



5 今後の課題

最後に今後の課題・予定を挙げる。先ず最初に述べたように、このアルゴリズムの実装は未だ成されていない。よって、このアルゴリズムを用いて、実際に印刷物中の行列を認識するアプリケーションの作成を行いたい。また、認識結果 (行列の構造) の保持を行うために行列を表すクラスの設計をする必要がある。それと同時に、実際の様々な行列のデータベースを作成し、アルゴリズムの改良、クラス設計に役立て、更には今回、対象外とした修飾記号にも対応できるようにしたい。

参考文献

[1] R. H. Anderson, *Systax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics*, Ph.D. Thesis, Div. of Eng. and Appl. Phys., Harvard Univ., Cambridge, Mssachusetts, 1968; also appears in summary from in *Interactive Systems for Experimental Applied Mathematics*, (M. Klerer and J. Reinfelds, eds.), pp. 436-459. Academic Press, New York, 1968.

[2] 岡本 正行, トワキョンド ムサフィリ ハシム, “周辺分布特徴を用いた数式構造認識”, 信学論, J78-D-II, No. 2, pp366-370(1995-2)

[3] 岡本 正行, 東 裕之, “記号レイアウトに注目した数式構造認識”, 信学論, J-78D-II, No. 3, pp474-482(1995-3)

[4] 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”, 信学技法, HCS97-8, pp7-14(1997-9)

[5] K.Inoue, R.Miyazaki, M.Suzuki, *Optical recognition of printed mathmatical documents*, Proceedings of the Third Asian Technology Conference in Mathematics, pp280-289, (1998-8)

Graduate School of Mathematics, Kyushu University 36, Fukuoka, 812-8581, Japan

E-mail address: kanahori@math.kyushu-u.ac.jp