

回答者の認知負荷を考慮した大学初等数学のための自動作問

江原 遥

Yo EHARA

静岡理科大学情報学部

Department of Informatics, Shizuoka Institute of Science and Technology

Email: ehara.yo@sist.ac.jp

あらまし：データサイエンス教育においては、線形代数や確率統計などの大学初等数学の習得が必要不可欠である。計算問題を解かせて新しい数学的概念を習得させるための計算問題の作問においては、例えば行列中の要素に桁の大きい数字が出てくるような、認知負荷の高い演算を防ぎたい。本稿では、逆行列や行列式などの計算で、問題中の行列の要素も答えも簡単な整数になるような問題を自動作問する手法を試作したので発表する。

1 はじめに

大学におけるデータサイエンス教育においては、「データ」の基本的な形である表やグラフの数学的表現である行列に関する数学、すなわち線形代数や、基礎的な統計教育が必要不可欠である。こうした大学の初等数学教育においては、講義を行っただけでは教育効果が不十分であり、実際に例題となる問題を解いて訓練すること、すなわち演習が必須となる。筆者は所属大学の情報学部で前期に線形代数/演習、後期に確率統計/演習を担当している（どちらも週2回×90分×15回）。

演習のためには、例題となる問題が必要になる。この際、どのような問題がよいだろうか？線形代数にしても確率統計にしても、大学の初等数学教育において求められる計算は、大半が分数や小数の四則演算のレベルである。演習問題で確認したいのは、学生が適切な立式ができるかどうかであり、計算力の高さを見たいわけではない。したがって、できるだけ少ない計算量で演習ができることが望ましい。

こうした「できるだけ少ない計算量」の問題は、一般に人手では作ることが難しい。例えば、次の2つの行列を用いて連立方程式を解く問題を見比べてほしい。

$$\begin{bmatrix} 3 & -2 & 2 \\ 5 & 4 & 1 \\ 4 & 5 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -4 \\ -1 \\ 5 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} 0 & -1 & -1 \\ 3 & 3 & -2 \\ -1 & -3 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \\ 2 \end{bmatrix} \quad (2)$$

どちらも単純に整数を係数とするガウスの消去法の演習問題である。しかし、式1の解は $x = -83/112$ 、 $y = -95/83$ 、 $z = 97/83$ とだいぶ手計算では複雑になるのに対して、式2の解は $x = 4$ 、 $y = -3$ 、 $z = 3$ と比較的簡単になる。実際の演習問題としては、明らかに式2の方が適切であろう。

従来、所与の演習問題を解くソフトウェアやWebサイトは¹など豊富に存在したが、式2のように計算が簡単になる演習問題を自動生成するソフトウェアやWebサイトは少なかった。そこで、本稿ではこのための手法を試作したのでこれについて説明する。

提案手法は、大量の問題を乱数を使って生成して、条件に合うものをフィルタリングする単純なアプローチを取った。行列群などを用いて条件に合う行列を同定するアプローチもあるが、そうした手法は取らなかった。その理由は、例えば、演習問題の計算においては、「答えが10進数で1桁の整数になる」など、数学的な代数構造の複雑さとは直接関係のない、慣習上の複雑さが計算の難しさに大きく影響しているため、計算の難しさに関する複雑な条件を入れにくいからである。提案手法のアプローチであれば、前述のような条件を簡単に入れ込むことができる。

本稿では、特にガウスの消去法による連立方程式の解法（逆行列の計算）に着目する。数値計算で使われるガウスの消去法をPythonのリスト機能を使って実装したものであるが、通常と実装と異なるのは、数値行列の要素を数式処理システム（Computer algebra system, CAS）上のオブジェクトを用いて実装している点異なる。これによって、手計算と同様、要素を分数として丸め誤差のない形で計算したり、要素に文字を用いることが可能となっている。Python言語上のライブラリとして無償でインストール可能な数式処理システム（Computer algebra system, CAS）ライブラリである **sympy** を用いて実装している。提案手法はGoogle Colaboratory上で実装されており、GmailアドレスとChromeブラウザさえあれば動作するようになっている。

本稿ではガウスの消去法に限定して解説するが、同様の手法は、少なくとも線形代数の講義については、行列式の計算など他にも広く応用できると思われる。本稿の貢献は以下のとおりである。

¹<https://keisan.casio.jp/>

- データサイエンス教育の現場では、大学の初等数学教育のために計算量の少ない演習問題を自動生成したいニーズがあることを報告し、新しい方向性のタスクを報告した。
- 実際に Web 上で簡単に問題を生成し、回答手順も生成する手法を提案・実装した。

2 実装

<http://pr4.work/0/jsisemath> に実装例を示す。紙面の制限のため、ソースコードについては URL を参照してほしい。Gmail アカウントさえあれば、Colaboratory を用いて動作させることができる。Colaboratory はブラウザ上で通信相手のサーバ上の Python 言語をインタラクティブに動作させることのできる Jupyter Notebook 環境の一種であるが、この実装では、Colaboratory 独自の機能は特に使っていない。したがって、計算機上に Jupyter Notebook 環境を用意することができれば、動作させることができる。

入力として、問題の行列の要素として認める整数の範囲と、乱数で生成する問題数を指定する。 $Ax = b$ において、行列 A の要素とベクトル b の要素が指定範囲の整数からランダムに選ぶ。問題を多数作成し、解 x が存在して簡単な整数値になるものだけを出力する。 A や b の要素として、`sympy` ライブラリの `Rational` オブジェクトを利用する事で、除算が出てきた場合でも分数が使われるようにする。連立方程式の解には、通常のガウスの消去法を用いるが、計算過程に分数が出てきた時点で不適とみなして次の問題をまたランダムに生成する。

出力例を図 1 に示す。最初の $\begin{bmatrix} -2 & -3 & -3 \\ -1 & -2 & -2 \\ -3 & -3 & 0 \end{bmatrix} x = \begin{bmatrix} -3 \\ -3 \\ -3 \end{bmatrix}$ は、次の問題が条件に当てはまることを示す。

$$\begin{bmatrix} -2 & -3 & -3 \\ -1 & -2 & -2 \\ -3 & -3 & 0 \end{bmatrix} x = \begin{bmatrix} -3 \\ -3 \\ -3 \end{bmatrix} \quad (3)$$

ガウスの消去法を用いているので、図 1 からわかるように、生成された問題から正解を導く手順も自動生成することができる。そのままテキスト形式でメモ帳などにコピー／貼り付けをすることによって学生にプロジェクターなどで回答手順を見せることができるように、回答手順は特殊な形式を使わずテキスト形式で表示されている。図 1 では、最初の問題に負の数が入っているが、 A や b の要素として許される整数の範囲を正の値に限定すれば、1桁の自然数だけから表される問題なども簡単に作ることができる。また、前述のソースコードでは、複数の b を与えることができるようになっている。そのため、例えば b として $[1, 0, 0]^T$ 、 $[0, 1, 0]^T$ 、 $[0, 0, 1]^T$ の 3本を同時に指定することによって、掃き出し法によって逆行列も整数になるような問題を自動生成することも容易である。10,000 個の問題を A や b の要素が -5 以上 4 以下になるようにランダムに作ったところ、33 個の問題が生成された。

```
([[-3, 12, -9]],
[[[-2, -3, -3, -3], [-1, -2, -2, -3], [-3, -1, 0, -3]]],
'====前進消去====',
'2行目を-1倍する。',
'2行目と1行目を入れ替える。',
'2行目に1行目の2倍を足す。',
'3行目に1行目の3倍を足す。',
[[[1, 2, 2, 3], [0, 1, 1, 3], [0, 5, 6, 6]],
'3行目に2行目の-5倍を足す。',
[[[1, 2, 2, 3], [0, 1, 1, 3], [0, 0, 1, -9]],
[[[1, 2, 2, 3], [0, 1, 1, 3], [0, 0, 1, -9]],
'====後退代入====',
'2行目に3行目の1倍を足す。',
[[[1, 2, 2, 3], [0, 1, 0, 12], [0, 0, 1, -9]],
'1行目に3行目の2倍を足す。',
[[[1, 2, 0, 21], [0, 1, 0, 12], [0, 0, 1, -9]],
'1行目に2行目の2倍を足す。',
[[[1, 0, 0, -3], [0, 1, 0, 12], [0, 0, 1, -9]]]]),
```

図 1: 出力例

3 関連研究

知る限り連立方程式の解を回答手順とともに作問する¹⁾が最も近い研究であると思われるが、目的と実装手段が異なる。¹⁾では計算時間を制御することが目的であったが、本研究では計算時間を制御すること自体は目的とはしておらず、受講者が適切な回答手順を理解できているかを確認するため、興味のない計算部分についてはなるべく簡単な問題を作成することが目的であった。実際、目的の違いのため、¹⁾では複素数や有理数を係数とする連立方程式の問題を作成する機能もあるが、本研究では、そうした問題を作成することは最初から目的の範囲外である。また、¹⁾では代表的な CAS の一つである Mathematica を用いているが、連立方程式の解法についてのみ主眼が置かれており、行列式など線形代数の他の問題も同様のアプローチで可能なことや一部が文字定数になっている問題の生成については述べられていない。実装の面では、提案手法では無償で利用できる Python 言語の機能だけを利用することによってより幅広い利用者が用いることができるようになった。

4 おわりに

本稿では、データサイエンスのためにガウスの消去法の練習問題を解く手法を提案した。今後の課題としては、計算の認知負荷をより精緻にモデル化することが挙げられる。

参考文献

- 森優貴, 雪田修一ほか. 計算量が制御された連立方程式の問題及び模範解答の自動生成. 第 76 回情報処理学会全国大会講演論文集, 第 2014 巻, pp. 831-833, 2014.