Moodle XML Question Generator for Python[1]

神戸大学・人間発達環境学研究科 長坂 耕作
Kosaku Nagasaka
Graduate School of Human Development and Environment,
Kobe University

1 はじめに

2017 年度開催の研究集会「数学ソフトウェアとその効果的教育利用に関する研究」では,『数式処理と学習管理システム – 静的評価の再評価 –』との表題にて，STACK 等の動的な評価でなく，静的な評価で十分ではないかと疑問を提示し，その仮説に基づき，Mathematica で Moodle 向けの多肢選択問題を生成するためのパッケージについて講演を行った [1]。また，同研究集会の 2018 年度においては,『Moodle KaTeX filter』との表題にて，Moodle にて静的な評価を多肢選択問題で行う際のタイムラグを問題視し，標準の MathJax に替えて，KaTeX を使用可能にする講演を行った [2]。

以上の成果に基づき，対面式講義におけるオンラインの演習問題を多肢選択問題で大規模に展開するシステムが構築可能となったが，実際の展開には大きな課題が残されていた。それは，線形代数などの数学の問題生成器を作成する環境として，Mathematica は確かに便利ではあるが，普及率や使用のし易さの観点からは最適と断言することが難しいということである。そこで本報告では，近年，幅広く利用が進む Jupyter Notebook 環境における Python の環境にて，Moodle 向けの多肢選択問題を生成するためのモジュールの開発について紹介する。

2 旧来の多肢選択問題の生成

2017 年度に発表 [1] した Mathematica において，Moodle 向けの多肢選択問題の自動生成を行うパッケージ「Moodle XML Questions Generator」については，CC BY-SA 4.0 のライセンスで，`https://wwwmain.h.kobe-u.ac.jp/~nagasaka/research/xml_quiz/` にて公開している（生成済みの線形代数の多肢選択問題も公開している）。このパッケージでは，予め問題毎にユーザー定義関数（QuestionText, QuestionGenerate, AnswerText, InCorrectGenerate など）を準備することで，あとはパッケージ関数が問題を自動生成する機能を提供していた。基本的な作成手順としては，次のようなものであった。

1. 問題の設計
2. 問題に必要となる関数の準備

[1]This work was supported by JSPS KAKENHI Grant Number 18K02941.

- 問題データの生成（個々の問題に必要な内部データの生成）

  ```
  QAData[問題名称，QuestionData[問題内部データ]，
         {AnswerData[点数，回答内部データ],...}]
  ```

- 正答データの生成（正答となる内部データの生成）
- 誤答データの生成（誤答となる内部データの生成）
- 問題文の生成（問題内部データを HTML に変換）
- 選択肢の生成（正答と誤答の内部データを HTML に変換）
- （Optional） フィードバックの生成
- （Optional） 生成データの検証

3. `GenerateQuestionSet` で XML を自動生成
4. Moodle の問題バンクにインポート

3 Python 版の機能

Python 版での開発にあたっては，実際の教育現場に投入した結果得られたフィードバックに基づいて，Mathematica 版には存在しなかった次のような機能を付与している。

- 問題毎に通し番号を付与可能
  - 大規模に展開した場合，教員/教師間での情報交換や，学生/生徒からの質問時などに問題を特定するためには，何らかの通し番号が必要なため。
- Jupyter Notebook 上での問題プレビュー
  - 問題生成器を作成する際，生成した問題の Moodle 上における見た目の確認が，Moodle に実際にインポートするまで不可能だと，作業が非効率なため。
- 問題文等に現れる文字列の置き換えが可能に
  - 中等教育では学習指導要領により用語の統一がある程度行われているが，高等教育では教科書毎・大学毎などでテクニカルタームが異なる場合が多い。このような場合にも問題の共有を可能とするには，文字列の置き換えが可能でなければならない。
- 生成した問題の内部形式での保存が可能に
  - 問題生成器が生成した問題を XML（Moodle の形式）に変換してしまうと，それまでの内部データが失われ，それ以上の編集が難しくなるため。

また，中等数学教育での多肢選択問題の自動生成も視野に入れ，新しいサンプル問題として，グラフィックスを生成するような問題生成器をPython版には追加した。加えて，多様なモジュールが存在する利点を活かすという観点から，青空文庫の文書からデータを取得して自然言語処理を行うことで問題を生成する例も作成した（本報告では，これらサンプルの詳細については割愛する）。

4　Python版の基本的な使い方

Python版は本報告執筆時点で未公開となっているが，現時点での仕様に基づきその概要を紹介する（未公開であるが，開発中のものを使ってみたい方は個別に連絡を）。

Python版のモジュールは，コアモジュールとして`moodle_xqg.core`，SymPyによる数式のLaTeX化などのラッパーをまとめたサブモジュールとして`moodle_xqg.qbank.common`などから構成され，通常は，自動生成器内で乱数生成や記号処理が必要なため，`random`や`sympy`などと共に使うことになる。その実際の問題生成までの流れは，概ね次の通り。

1. `moodle_xqg.core.Question`の小クラスとして生成器を定義
    - 必要なメソッド（`Question`は抽象クラスやインターフェイス的な性格）
      `question_generate`, `correct_answers_generate`,
      `incorrect_answers_generate`, `question_text`, `answer_text`
2. 問題生成器のインスタンスを生成
    - 生成器毎の細かな挙動は，インスタンス毎に調整
3. 実際に問題を大量に生成
    - `moodle_xqg.core.Quizzes`のインスタンスとして生成される
    - `list`メソッドで箇条書きに画面出力
    - `preview`メソッドで実際の選択画面として出力
    - 最終的に確認後，`save`メソッドでXMLなどでファイル出力

5　今後の課題

本報告時点（研究集会開催期間）においても，本報告執筆時点においても，Python版の基本的な部分は完成している。非公開となっているのは，実際に，線形代数の問題生成器をMathematica版から移植することにより，Python版の完成度を高めてから公開するためである。以下で述べる今後の課題は，この作業中に発案されたものとなる。

線形代数の問題生成器をMathematica版からPython版に移植するにあたり，共通する関数などをクラスとして集約する作業を行った（Mathematicaには，クラスという概念が基本的には存在しない）。この作業の結果，少なくとも，次の2つのクラス（`LinearSpace`と`LinearSubSpace`）を定義するに至った。

LinearSpace：線形空間を表すクラス

- 問題を超えて共通の線形空間の生成が必要
  - 解空間（同次線形方程式）＋解集合（非同次）
  - カーネル（零空間）とイメージ（行空間，列空間）
- 誤答生成には，柔軟な表現や基底の変換などが必要
  - 生成系から基底への変換・逆変換（簡約化含む）
  - 直交空間の生成など

LinearSubSpace：線形空間の部分空間を表すクラス

- 様々な同型な部分空間を行き来する変換など

しかしながら，これら 2 つのクラスは本来ならば区別する必要はないが，様々な事情から統合できていない。理由としては，問題生成器により，問題の難易度に直結する簡易的な生成方法などが異なり，単純な統合が難しいことなどが挙げられる。この状況は，今後の課題として，体上の加群としての線形空間の構造ではなく，問題生成器を前提した場合の線形空間の構造の研究の重要性を示唆している。

例えば，計算機代数を研究する立場からすれば，一般の線形空間の問題は数行列の問題に帰着されるため，数行列を対象とした簡約（reduced row echelon form）などの各種数行列計算が出来れば十分と言え，同型な線形空間の間の見た目の表現の変更などは本質的な問題とはならない。

一方で，本取組のように，各種の問題生成器を作り易くする立場からすれば，例えば，同じ線形空間を同次線形方程式の解空間として解釈した場合と，幾つかの冗長なベクトルを生成系に持つ部分空間として解釈した場合とで，問題の見た目から受ける難易度や実際の基底計算などの計算問題としての難易度などが大きく異なるのは困ることになる。本質的には同じ線形空間であっても，多様な切り口による様々な問題設定において，難易度のコントロールが容易であるような構造が求められることになる。これは，多肢選択問題の自動生成を大規模に線形代数で展開する上で，今後早急に解決すべき課題の 1 つとして考えられる。

参考文献

[1] 長坂耕作. 数式処理と学習管理システム - 静的評価の再評価 -. 研究集会 数学ソフトウェアとその効果的教育利用に関する研究 (2017 年 9 月). **数理解析研究所講究録**, 2067:160–169. 2018.

[2] 長坂耕作. Moodle KaTeX filter. 研究集会 数学ソフトウェアとその効果的教育利用に関する研究 (2018 年 8 月). **数理解析研究所講究録**, 2105:106–108. 2019.