Educational technology in the large mathematics classroom

Duane Nykamp
University of Minnesota
School of Mathematics
Sage Days, August 2017

Motivation for involving technology in the classroom

- Active learning = good
- Large classes = inevitable
- Instructor interaction = critical
- Key instructor role = coach

How can we best use technology to maximize the effectiveness of instructor coaching interactions in a large active learning mathematics class?

In person or via technology?

	In person	Via technology
Content delivery	Tailor based on students' reaction	Available any time
Assessment	Concepts, high-level thinking	Computations
Feedback	Nuanced, individualized feedback	Immediate feedback to each student
Guidance	Diagnose trouble spots	Indicate standard routes to common problems
Grading	Open ended, free responses	Easily allow multiple attempts

Strategy we've adopted

Developed flipped College Algebra, Precalculus, and Biocalculus courses with a large educational technology component.

- Content delivery, videos and/or readings, at home
- Projects, exploration, worksheets, problem-solving, communication skills in class

Roles of educational technology

- Homework assignments
- Computer-graded worksheets
- Online quizzes
- Computer-graded exams
- Generate hand-graded exams

Developed educational technology

Developed our own educational technology systems

- MOLS (Minnesota Online Learning System)
  - Based on Mathematica
  - Used for active courses, review materials, and placement exams
- Math Insight (mathinsight.org)
  - Based on python (sympy)
  - Integrates online text, videos, interactive applets, interactive text, computer-graded problems

MOLS (Minnesota Online Learning System)

Demonstration

Math Insight

mathinsight.org

Examples

- Helping students translate between different graphical representations
- Interaction between interactives and text
- Potential for richer graphical interaction

Challenges in answer validation, feedback, and partial credit

Basic question: if I have two similar expressions a and a', how can I

1. determine where they are different
2. provide feedback or hints about their difference
3. provide mechanisms for assigning partial credit
4. provide customized instructions or solutions based on errors

Simple example

Enter $x + y + 2.8821$

Exact:

Rounding:

Sign errors:

Submit

Catch errors in constant terms or factors?

Problems/ambiguities when extending to numerical errors.

- Type of error depends on mathematical representation
- Give useful/understandable feedback?

Enter $3y + e^{5x+1} + 2$

Submit

Appropriate comparison depends on context.

Appropriate feedback and partial credit for deviation from the correct expression $a = \frac{4}{3}x^3$ may depend on the problem. Common errors would be different, for example, in these cases:

- $$\frac{\mathrm{d}}{\mathrm{d}x}\frac{x^4}{3}$$
- $\displaystyle\int 4x^2 dx$ (ignoring constant)
- $\dfrac{4x^3 - 8x^4}{3 - 6x}$ if $x \neq \frac{1}{2}$

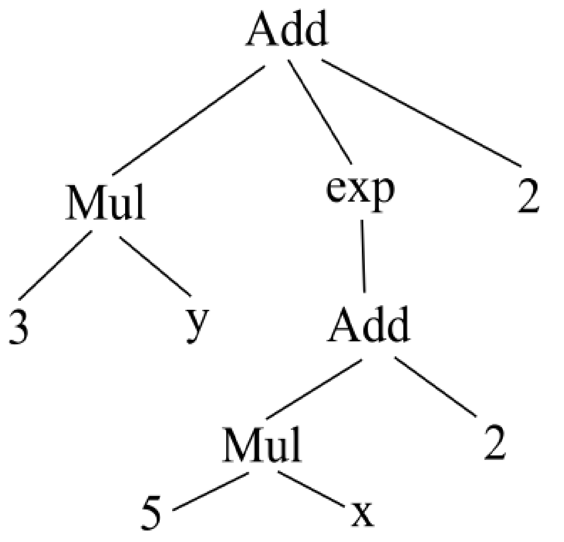
An effective feedback and partial credit system should reflect these errors.

Beyond the expression tree

Previous approach for comparing expressions a and a' was based on manipulating their expression trees.

Expression trees (at least when simplified or evaluated) contain insufficient information to determine effective feedback and partial credit.

$$3y + e^{5x+2} + 2$$



Question: Can one construct expressions with richer metadata and exploit the structure to provide better feedback and partial credit?

Unevaluated expressions

Idea: base approach on unevaluated expressions.

Rather than coding an answer as an expression $a = \frac{4}{3}x^3$, one retains the operations used to obtain the answer. The previous examples could be written as:

- `diff(x^4/3, x)`
- `integrate(4*x^2, x)`
- `simplify_rational((4*x^3-8*x^4)/(3-6*x))`

Expression trees involving unevaluated operations could underlie more sophisticated answer validation algorithms.

Operations with errors

Challenge: identify common errors and program “corrupted” functions that capture the error.

For example, an `integrate` function that integrates x^n as $\frac{1}{n}x^{n+1}$.

When substitute combinations of corrupted functions into the unevaluated expression tree of the answer, obtain a family corrupted expression trees representing different combinations of errors.

In answer validation,

1. Attempt to match expression tree of a user's response to a corrupted expression tree.
2. If match found, give appropriate feedback and/or partial credit.

Example

All answers are $\frac{4x^3}{3}$.

1. $\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{x^4}{3}\right) =$
2. $\int 4x^2 dx =$

Submit

Possible extensions of corrupted tree approach

Could extend approach to check for other ways of corrupting expression tree

Examples

- Messing up order of operations
- Omitting steps (missing parts of tree)
- Correct answer based on user's previous incorrect response

Advantages of corrupted tree approach

- Intricacies hidden from question author
  - Author simply writes correct answer as expression tree
  - System computes common corrupted trees behind the scenes
  - Automatic partial credit and/or feedback

- Straightforward approach to give authors more control
  - Sensible defaults for generic expression tree
  - Author could add additional markup (attributes) to segments of the expression tree to change behavior

Challenges to corrupted tree approach

- Identifying “common errors” and appropriate feedback
- Determining what constitutes a match to a corrupted tree
- Number of corrupted trees to consider increases exponentially with each point of corruption