Towards Collaborative Intelligent Tutors

Ya’akov (Kobi) Gal, Elif Yamangil, Stuart Shieber, Andee Rubin, Barbara Grosz

The Goal

Automatic recognition of users’ interaction with pedagogical software.
A first step towards a collaborative system that supports
- students in their problem-solving,
- teachers in their analysis of students’ performance.

Outline

The TinkerPlots domain
- Recognition of users’ activities
Automatic Plan recognition in
- Artificial intelligence
- TinkerPlots and pedagogical software
Empirical Methodology
- Algorithm description
- Experiment and analysis

TinkerPlots

A construction kit for generating and analyzing data.
Used by 11 and 12 year old middle-school students in Holyoke MA and TN.
The Two Dice Problem

We rolled two dice and kept track of how often we obtain the sums 2 through 12.

For example, when one die came up 5 and the other die came up 3, we recorded their sum as 8.

Does the sum 11 come up more often than 12? Why?

A Student’s Possible Model

Each roll of the dice is an event consisting of two values.

• (1,1), (2,2), ...

There are 36 equally likely events.

Two events give the sum of 11

• (5,6), (6,5)

One event gives 12

• (6,6)

Hypothesis:

• (5,6), (6,5) are distinct events

• 11 will occur more often than 12.

What Teachers Want...

How do students construct TinkerPlots models?

How do they explore, interleave between actions sequences?

How do they use the models to analyze data?

What type of mistakes do they make?

Lab Session “Chaos”
Possible Approaches

Taking screen snapshots
- Obscures students’ order of activities, their exploration, their mistakes.

Goal recognition in ITS [Anderson, Conatti, Corbett, Roll, VanLehn]
- Predicting users’ future actions.
- Tutor is an active participant in interaction.
- Do not construct a complete plan of user’s activities over time.

Basic Actions

Complex Actions

Recipes

Basic actions are **rudimentary**. Complex actions are **abstract**. Composed of other complex and basic actions.

A **recipe** for a complex action includes a series of (complex and basic) sub-actions for completing the complex action;
Planning

A plan for a complex action
• set of recipes chosen towards completing the action.
• Represents users’ activities with TinkerPlots.
The act of “doing” these actions realizes the plan.
Basic level actions are
• linear (temporally ordered)
• observed
Plans are
• hierarchies of recipes
• unobserved

Plan Recognition in AI

Infer the plan of the agent given observations and a data base of recipes.

Observations are basic actions.
Applications for Plan Recognition

Speech recognition systems
Transportation routines (e.g., goal detection via GPS)
Opponent modeling (e.g., email communication between terrorists)

Plan Recognition in TinkerPlots

Input
- A sequence of basic level TinkerPlots actions,
- A data base of recipes.

Output
- The plan representing users’ activities in TinkerPlots

Traditional Assumptions

Goal oriented agent; no mistakes,
Outputs a single plan hierarchy,
Ideal recipes,
No interleaving of actions,
Actions appear in fixed order.

Is this the way people use computer software?
Advantages of TinkerPlots

Domain

Open-ended interface. Users may
• interleave between TinkerPlots activities,
• perform actions in “loose” order
• make mistakes

Free Order

Add Sampler
Set repetitions
Run Sampler
Add Plot

Roll one
Die

Roll Two
Dice

Solve
Two Dice
Problem

Fit Data to
Plot

Create Two Dice

Roll Two Dice

Fit Data to Plot

Solve Two Dice
Problem
Naive Solution

Naive Solution:
- Define separate recipes for each possible action combination.
But the number of possible permutations is factorial in the number of actions!

TinkerPlots Insights

Don’t need to consider action combinations that are disallowed by TinkerPlots itself.

Temporal Constraints

Actions for recipes should be selected non-contiguously.
TinkerPlots Insights

Later actions may be more salient than earlier actions.

Algorithm

Input: A set of TinkerPlots basic actions, a set of recipes. Traverse the action sequence in reverse order, Choose a sub-set of (possibly non-contiguous) actions to complete a recipe for a complex action from the data base. “Push” the complex action to the action sequence. “Pop” the constituent actions off of the action sequence. Reiterate and traverse the new action set.

Output: A set of basic and complex level actions representing the user’s plan.
Empirical Evaluation

Used 8 subjects (middle school students and adults)

Subjects
• saw a five minute video about TinkerPlots.
• Given identical half-hour tutorial.
• Asked to solve one of two types of problems
  • Two Dice Problem,
  • ROSA Problem.

Domain expert
• Constructed recipes,
• Observed subjects’ interaction with TinkerPlots.

Results

A plan is **correct** if the plan found by algorithm matched the judgement of a human expert.

Algorithm was able to recognize all interactions but one.
• Subject solved the two-dice problem twice.
• Algorithm only recognized one of those instances.
• Recognizing all possible ways to fulfill recipes is hard.

Summary

Rich pedagogical software environments
• Facilitate students’ learning of math concepts.
• Difficult to recognize users’ activities

AI techniques can facilitate the solutions of these problems.

Future Work

Presentation to teachers
• How best to present aggregate and individual performance,
• Combine state-based vs. plan-based representations.

Machine-generated support
• Tailor individual,
• Guide student towards better learning experience,
• Learn recipes from data.
Acknowledgements

Swapna Reddy
Craig Miller
Cliff Conold