Research Statement

Austin Cory Bart

Introduction

My research stands at the intersection of Computer Science Education, Program Analysis, Data Science, and Learning Science. In effect, I live in two worlds: one driven by algorithms and data, and the other driven by student models and learning theory. My ultimate goal is to maximize learners' potential while minimizing instructor effort, at scale. I believe the combination of advanced computing techniques and modern learning science research is the secret to achieving this goal.

My position at the intersection of computing and learning allows me to address one of the major computer science education challenges. As computing classes grow in size and start catering to wider audiences, we are forced to confront the inadequacies of our curricula to grapple with large audiences of diverse learners. With the right support, many struggling students could be successful; however, the existing educational infrastructure does not give learners the resources and scaffolds they need. To serve a broad computing population, we need better tools and pedagogical practices. I have been involved in a number of projects to serve large computing populations with new tools and techniques. It is important that new technology and curriculum be grounded in relevant learning theory and informed by study of the learners' needs.

Addressing these challenges requires me to secure external research funding. I have had success during graduate school in obtaining NSF funding for my research projects related to Computing Education. In my second year, I was awarded the NSF Graduate Research Fellowship to complete the work that would become my dissertation. In my final year, I assisted in the authoring of an NSF-funded grant to advance my dissertation work further and to scale it to multiple universities; this grant now supports my work at Virginia Tech as a Visiting Assistant Professor. I look forward to future NSF funding opportunities, as the importance of computing education becomes more visible to policy-makers.

Current Research

My research so far has been largely focused on introductory computing experiences with students, leading to the development and evaluation of software that supports learners' motivation and cognition. This software has been deployed in multiple classrooms and the related research findings have been published in a number of conferences (including a best paper award at SIGCSE 2017) and CS Ed journals.

Pedagogical Datasets

The focus of my dissertation was the contextualization of introductory computing experiences, in order to motivate non-majors who have concerns about the usefulness of the computational material. This led to the development of two projects: RealTimeWeb and the CORGIS project. The RealTimeWeb project was a small collection of datasets that made web-based data available despite hazards such as network failure, changing APIs, and poor web connections. The CORGIS project (the Collection of Really Great and Interesting dataSets) makes real-world datasets available to students in formats for a wide variety of introductory languages (Python, Java, and Racket). The current collection has 40+ diverse datasets drawn from subjects such as history, health, geology, art, and literature. Research studies conducted with these datasets in real classrooms found a positive impact on student motivation, even compared to other contexts. My work on the CORGIS project led to a Best CS Ed Research Paper award at SIGCSE'17 and a followup paper at SIGCSE'18, and both projects have had workshops and conference talks.

Dual Block/Text Guiding Programming Environment

Introductory students often struggle with the basics of syntax, even in relatively simple languages like Python. I developed BlockPy to explore how a dual block/text interface can support novice learners. BlockPy uses a combination of the Blockly interface library and the Skulpt JavaScript Python execution engine to allow mutual language translation between the block and text interface using AST traversal. Although BlockPy was successful in minimizing students' struggles with syntax, it became clear that students needed more support with program development. Therefore, BlockPy was extended with immediate, targeted feedback to students. Student programs in introductory courses are short, have certain expected characteristics, and require only a subset of the language features. These characteristics make it easy to apply analyses like Abstract Interpretation to conduct simple static program analysis, gather def/use data and limited type inferencing, and provide enhanced feedback on students' work. Studies on this environment have found it to be effective at supporting learners at scale, and we are now developing new interfaces to support both learners and instructors.

Future Work

As I look to the future of my research, I am particularly interested in a number of projects that expand the computational, analytical, and pedagogical methods that I have used in the past.

Student Program Analysis

The Immediate Feedback support in BlockPy has been very effective at helping students correct mistakes in their programs - however, there are still limitations in the extent of the feedback that the environment can produce. I am interested in how advanced program analysis techniques (e.g., symbolic execution, type inferencing) can be used to infer more extensive information about mistakes in student programs, and how we can best present feedback that overcomes misconceptions. Techniques like canonical program representation could aid in clustering student submissions to identify the most common recurring student errors. There are even more specific types of analyses that could be applied, such as variable role inferencing to highlight the semantic meaning of the variable. Ultimately, I would like to create systems that, with minimal effort, will empower instructors to create very sophisticated immediate feedback for programming assignments in semi-automated ways. This work would be supported by collaborations with researchers studying programming languages and software engineering, and also learning scientists who have experience in designing feedback for students.

Integrated Worked Examples

Although immediate feedback can be effective at supporting learners in the middle of a problem, I find there are usually a small but non-trivial percentage of students who struggle with getting started on problems. Either because they are unclear about the assignment, unclear about the goal, or unclear about what methods to use, these students encounter a blank canvas and seem to freeze up. Typically, these students require interventions involving considerable time and attention. According to the literature, well-formed and documented demonstrations of solving related problems named Worked Examples are one effective tool for helping learners in these situations. I want to study how collections of high-quality worked examples can be embedded directly within programming environments such as BlockPy in a tutorial mode, whether they support learners efficiently and as effectively as human tutors, and what tools are necessary to support instructors in constructing these Worked Examples. For this project, I foresee collaborating with educational researchers and researchers in human-computer interaction.

Data Analytics

What data do instructors need to support their learners, manage their staff, and revise their curriculum? How can we present this data to instructors to most efficiently? Modern Learning Management Systems provide APIs that allow us to extract data about learners, curricula, and staff for Learning Analytics. In prior courses, I have collected data from many sources, such as keystroke logs from coding environments or multiple submissions in mastery learning-style multiple choice quizzes. With a properly aligned curriculum that is carefully instrumented, instructors can gain deep insight into their course. There are many interesting potential objectives beyond improving course efficiency - for example, measuring impact on student learning and self-regulation, or identifying the most common misconceptions that linger throughout a course. The data collection and presentation is only the start however - this project would go the next step and also help instructors and course staff directly interact with students, making data-driven suggestions about specific interventions that could help students. This cohesive data platform would be an engine for improving courses across disciplines and answering course-specific research questions, and would involve collaborations with statisticians and data scientists.

Learning Experience Design

Even as computing education struggles to cope with the demands from an ever broadening audience and their educational needs, the community of computing educators struggles to obtain and maintain sufficient high-quality curricular materials for the courses they teach. How can we package learning experiences for reusability and replicability? How do we train instructors and their staff to quickly integrate the content, technical, and pedagogical knowledge required to leverage new learning experiences? This goes beyond prior research on Open Educational Resources, to provide an *entire* curriculum package "in a box" - not just slides or problems, but flexible sets of learning objectives, students' misconceptions, instructor interventions, staff training materials, and more. To accomplish this, we need to apply systematic Instructional Design, concrete ways to align learning objectives, assessments, and instructional materials.

Designed Data

Although the CORGIS project demonstrated that a high-quality collection of datasets could be a powerful motivational tool for contextualizing introductory computing experiences, a major limiting factor was the preparation and maintenance of the pedagogical datasets. Many fields do not make public, high-quality datasets available, because of dangers to vulnerable populations, or for competitive advantage, or out of pragmatics, or for any of a dozen other reasons. Simultaneously, the regular need for maintenance of the datasets inhibits future datasets development and growth. To overcome these limitations, I am interested in exploring how artificial pedagogical datasets could be generated using data fabrication techniques. These datasets would reflect real-world characteristics of data that it was drawn from, while being free of the complexity that real-world data so often brings. I am curious to what extent the authenticity of a dataset be sacrificed for the sake of simplifying the exploration of the data. I am also curious if an an ecosystem of artificial datasets could be generated that reflect a centralized narrative, creating a rich and entertaining experience for learners that could motivate them in ways that real data could not. This project would require collaboration with researchers across disciplines in the social sciences, humanities, and creative arts.

Conclusion

I believe that computing education is at an exciting inflection point that necessitates drawing new methods from both computational and educational domains, synthesizing the two in ways that will allow us to support more students according to their individual needs. I look forward to projects that incorporate innovative software and pedagogical methods that will scale our impact to broader communities, extending my prior work in novel ways.