Academic statement, Shashank Srikant

The surge of innovations owing to machine learning and data science is a recent phenomenon. Its impact today is mostly on problems in text, images and sound. I am interested in exploring a rather different, yet equally prevalent domain - computer programs. I'm most excited to build a robust framework to examine computer programs statistically. Having such statistical models can help design a number of technologies like scalably translating one language to another, topic modeling in programs and associating code reviews written in a natural language to constructs in a programming language. I believe these will seem obvious tools to have in the days to come, like Google's Maps and Translate are today. Such models will help us reason about programs better, and will eventually make programming easy to learn, understand and teach. This work sits at the intersection of program analysis, machine learning and HCI.

Program grading using ML

My research in data-driven program analysis began at Aspiring Minds' research lab. Aspiring Minds uses technology to quantify skills to efficiently map people to the right jobs. In 2012, I worked with Varun Aggarwal (CTO, SM '07 EECS, MIT) to build Automata, a tool to holistically grade computer programs [1]. A critical component was to determine whether a program was semantically `close' to a correct answer. This was traditionally done either by evaluating test-suites or by solving constraint equations using SAT solvers. The former is an error-prone metric while the latter infeasible for real-time systems. We designed a grammar for generating features from the abstract syntax trees of programs. They captured data and control dependencies between variables, which we showed to provide significant additional information. Supervised models learnt on these features had a correlation of ~0.9 with expert raters, rivaling inter-rater correlations. We published these strong results at KDD 2014 [2]. We also launched a software product using this technology, an effort I led from scratch. This product is used by tech-giants like Amazon and Baidu today and has evaluated more than two million students. We also authored a position-paper describing the broader framework of applying machine learning to formative assessments [3]. I designed various other tools which solved related problems in program evaluation - clustering test cases in a test suite to detect anomalies [4], empirically detecting the time complexity of programs and preparing style guidelines for programs by mining code repositories.

Question independent program grading

A particular bottleneck in our KDD '14 work was in building predictive models for every programming question we designed. It is a time consuming and human-intensive effort. Extending our work, we conceived a way to transform program features from each question onto a question-independent space. This question-independent space quantified a program's correctness without needing labeled data. Such a metric allowed us to compare programs attempting different questions on the same scale. We showed that supervised models learnt on these transformed features were able to predict as well as those learnt on question-specific features [5]. This scalable system has since reduced our content development effort by 80%. We also illustrated how this framework can address similar, yet unsolved problems in formative assessments. While solving this problem, I also recognized how it was not entirely a problem in transfer learning theory. Though the questions shared the same feature space, the overall learning task changed drastically between problems. I'm keen on investigating how the theory can accommodate this class of problems.

Models for job selection & NLP

Besides program analysis, I have worked on other problems in ML and NLP. I helped design classifiers to signal employability based on test scores. We wanted models which were piece-wise monotonic and interpretable. We successfully designed a suite of linear models which solved a non-convex cost-function. We showed that they outperformed traditional linear classifiers and provided a rich interpretation of how people can be hired [6]. In other work, I was exposed to challenges in NLP when I mapped people's job aspirations to relevant educational resources on the web. I dealt with very noisy datasets and had to engineer systems to make the learning techniques work in our context. We now have this as an internal system with a 70-70 precision-recall to recommend a host of services based on aspiration information provided by end-users [7].

Initiatives

I feel strongly about research outreach. It is our responsibility as researchers to highlight the role of techniques like ML as important cross-disciplinary tools. To this end, I co-organized international workshops (one at KDD 2014 and another at ICDM 2015) to get the machine learning and education community to exchange notes on the state of the art [8]. I also co-founded Data science for kids [9], a fun initiative to teach school kids the core ideas of data science using engaging and novel hands-on exercises. The design principles behind this initiative have been accepted for publication at SIGCSE 2017, having received very positive reviews [10]. I have enjoyed conveying my research to a senior audience as well. I have given invited lectures at graduate-level courses on program analysis at IIT Madras and other technology institutes across India. I have also talked about my work at Xerox Research and Microsoft Research, India. To track the latest goings on in ML in India, I helped put together http://ml-india.org.

Looking forward, I would like to work in the space of data-driven program analysis. Given my extensive training in both, machine learning and program analysis, I'm confident of solving problems in this area. I believe I'll find a great fit in working with Prof. William Cohen and Prof. Umut Acar from computer science given their focus on cross-disciplinary research in machine learning and program analysis. Likewise, ISR promises to be a great place to investigate this area further. Specifically, professors Jonathan Aldrich, Christian Kastner, Claire Le Goues and Bill Scherlis' interests in topics at the intersection of machine learning and program analysis driven systems is an ideal match with my research interest. From the HCI institute, Prof. Ken Koedinger's research on automated tools for programmer feedback and evaluation is strongly aligned to my interests. CMU promises to provide me a stimulating environment to continue my education and research in a field that is yet to be fully explored. I eagerly look forward to it!

References

[1] http://spectrum.ieee.org/at-work/education/moocs-teach-ai-to-grade-computer-programs-as-a-human-would
[2] A system to grade computer programs using machine learning, Srikant S., Aggarwal V., KDD 2014
[3] Principles for using machine learning in the assessment of open response items: Programming assessment as a case study, Aggarwal V., Srikant S. and Shashidhar, V., Workshop on Data Driven Education, NIPS 2013
[4] http://research.aspiringminds.com/test-case-analytics/

[5] Question independent grading using machine learning: A case of computer program grading, Singh G., Srikant S., Aggarwal V., KDD 2016
[6] Learning Models for Personalized Educational Feedback and Job Selection, Shashidhar V., Srikant S., Aggarwal V., Workshop on ML for Education, ICML 2015
[7] http://research.aspiringminds.com/large-multi-class-classification/
[8] http://aspiringminds.com/assess/2015
[9] http://datasciencekids.org
[10] Introducing data science to school kids, Srikant S., Aggarwal V., SIGCSE 2017 (To appear)

Other project details on webpage: https://shashank-srikant.github.io/