Teaching Statement

KMA Solaiman

1 Teaching Philosophy

Accesible Learning My teaching philosophy is based upon active learning and student well-being. Student well-being in a classroom can depend on their sense of autonomy, achievment, and participation. For example, in one of the courses I taught CS536, the students were provided a choice for their project topics from reproducible papers, novel idea implementation, or their ongoing projects. Also, for the final presentation deliverable they were asked to choose from multiple modalities i.e., video presentation, demo, or blog write-up. These choices gave the students a sense of autonomy. Satisfying these needs could intrinsically motivate the students to actively participate and grow in the classroom.

Relationship with the Students In the beginning of my classes, I make a conscious effort to know the students backgrounds, and their preferred outcomes from that course. This helps me understand them better, formulate relevant examples during the teaching, and allow me to adjust the course outline accordingly. *"If you cannot explain it simply, you do not understand it well enough."* - I truely believe that, and want that as a final outcome for my students. To that end, I ask for *frequent feedbacks* from my students to learn more about their understanding and thought processes, as well as identify any alternate conceptions students may have that need to be addressed.

Growth-focused Course Design (i) *Course Materials:* I prefer to design courses in a way that gives the students the ability to measure their own progress. The courses consist of both the foundational CS knowledge and the application of the concepts to problem solving. Depending on the audience type, the course may have varying style of materials. For students aspiring industry positions it is imperative that they can apply the academic knowledge to solve real-life problems whereas for the students pursuing research careers it is important to be able to come up with solutions to new problems from the existing knowledge. I grew very fond of the concept of *learning by doing* from my own courses and prefer to utilize this. Allowing students to do different types of written or programming assignments along side the relevant lectures bolster their understanding of the theoretical knowledge. (ii) *Assesment:* During grading, I always design rubrics that are very detailed, have lower stakes, and provide significant feedback to the students so that they can understand what mistakes they have made and can learn from it. Along with testing the fundamental and applied knowledge, I frequently add problems during the coursework that make them think on a deeper level.

2 Teaching Experience

My teaching experience consists of both hands-on teaching for 7 years, dating back to 2014, and taking training courses such as 'Effective teaching in CS', or 'Foundations of College Teaching'. The graduate TA training after COVID-19 has helped me get more familiarized with online teaching while learning about essential tools such as Brightspace, Campuswire, Piazza, Gradescope, Vocareum, etc.

My first teaching experience was in Summer'14 when I started teaching *Computer Graphics* to undergraduates in Bangladesh. This was my first experience dealing with students from different educational backgrounds and building their core knowledge about computer science through courses such as, *Programming Languages* and *Data Structure*. My second teaching experience was on a much broader scale at another university (AUST) in Bangladesh teaching *Network Programming* to undergraduates with a class size of 145. I continued teaching large classes of *Database, Structured Programming Language,* and *Software Programming* till Spring'16 at AUST. My duties included teaching the class, designing labs and lecture materials, conducting lab sessions, and grading. Through guiding these large classes I learned how to advise, guide, and evaluate large classes in a balanced manner. The satisfaction and joy I felt seeing some of the software programming projects by freshmen bolstered my choice of being a teacher.

I joined the Ph.D. program in Computer Science at Purdue University in Fall 2016 and became a teaching assistant for the *object-oriented programming* course for undergraduates. During 2016-22, I was a teaching assistant for two graduate-level courses along with three undergraduate courses. The biggest change I faced when I started teaching at Purdue was the multicultural and diversified international student body. In Purdue, my responsibilities as a GTA included designing, testing, and grading programming assignments, projects, and written homework, teaching labs and PSO sessions, assisting in creating and grading exams, and advising students during office hours and in online forums. Through this process, I have realized how differently students at senior and junior levels learn and communicate with instructors. For CS180 and CS251, I was responsible for overseeing undergraduate TAs, whereas for graduate classes I often had to discuss with students fundamental research questions. My goal for the weekly PSO sessions was to help students utilize the concepts learned in the lectures for their assignments with an in-depth understanding. As soon as an assignment was released, I went through the logic behind it and how did the concepts build up to the final assignment. During the grading, I tried my best to leave detailed feedback of the issues in their implementation, so they can learn from their mistakes. Mental health of the students is very important to me, and I encourage and practice empathy in my class, which includes being aware of their hesitancy to ask questions, or being inquisitve about their learning process. Most recently one of my students in network programming class felt comfortable enough to talk to me about his anxiety, and I have tried my best to accomodate and comfort him with Purdue's ongoing support for mental health management.

I have guest lectured on *situational knowledge, knowledge graphs, and multimodal information retrieval* where I talked about *cross correlation learning, metric learning, decoder-encoder, and attention networks*. The lectures involved *feature extraction from multiple modalities, graph embedding,* and *graph matching techniques*. I have always enjoyed public speaking and that has helped me to deliver presentations at a large scale numerous times during my Ph.D. I have presented research from me and my colleagues at Northrup Grumman Corporation (NGC) Review Meetings, NGC Techfest (with 100+ attendants), JPL Nasa, and Darpa Review Meetings.

Mentorship: During my time in AUST, I mentored four senior year students for their thesis project, and I actively participated in grooming students for *'Competitive Programming Competition'*. During the PhD, I have mentored multiple M.Sc. and undergraduate students for independent research courses.

3 Teaching Plans

I am excited to teach both undergraduate and graduate level students. I am comfortable teaching databases, networks, compilers, information retrieval, data management systems (SQL and NoSQL systems), and machine learning courses to both undergraduate and graduate students. I know the differences between them from my own experience of teaching at both levels. For undergraduate students, I would be happy to teach structured and object-oriented programming, data structure, theory of computation, and computer graphics. Specifically, for graudate students I can offer advanced information retrieval, natural language processing, distributed database systems, or data mining. Moreover, I have plans to offer seminar courses related to multimodal information retrieval and adaptable and explainable AI to students with interest in research. These courses will be based on recently published papers in top conferences in machine learning, information retrieval, and XAI conferences. The course projects will be designed in a way that can lead to publications and can mimic peer-reviewing.

Besides that, I would also like to develop a new course *"Applied Machine Learning for Open World Systems"* to extend the current curriculum based on my research experience. Tentative topics would include data cleaning, handling lack of annotations, scalabitlity, weakly supervised learning, multimodal information retrieval and feature extraction, intrinsic and extrinsic complexity of system domains, and case study of real world systems. This would also include detection, adaptation, and difficulty analysis of novelties in learning algorithms.