Teaching Statement

Nipun Batra
February 25, 2018

Throughout my life, I have been fortunate to have worked with and learnt from inspiring and influential teachers and mentors. I have always strived to "pay it forward". Thus, imparting education and student interaction were one of my primary motivators towards an academic career, which I strongly believe is one of the best ways I can contribute to the society.

TEACHING METHODOLOGY

Based on my observations when I was as a student, and my teaching and organisational experiences, here are some methodologies that I would like to incorporate into my teaching.

PROGRAMMATICALLY UNDERSTANDING CONCEPTS

Prof. Ashwin's (at IIITD) statement during a lecture on statistics that everything boils down to additions and subtractions left a significant impact on the way I learn. I took a programmatic approach towards understanding various statistical and machine learning concepts powered by visualisation. The additional insights I obtained in the process more than compensated for the time invested in the exercise. I maintain an active blog where I present many of my understandings gained in these ways. Some of these posts[1] have been used in courses at places like UIUC, and some of the articles occur on the first page of a google search for the concept: programmatically understanding dynamic time warping[2], programmatically understanding expectation maxmisation[3], dummies guide to the Fourier transform[4], among many. I would like to present my students with a similar learning experience, where the focus is on minimising the problem to a bare minimum, implementing it, visualising and tweaking to obtain additional insights.

RELATING TO THE REAL WORD

I have felt that if I can relate to some concept in my life, I have an easier time understanding it. For example, relating insertion sort to arranging a pack of cards; or explaining algorithms via problems the students can relate to, like hostel allocation, etc. Thus, for many of the topics that I teach, I plan to have real-world examples that the students can remember, and have an easier time understanding. In my prior teaching roles as a teaching assistant and PhD bootcamp organiser, I put significant effort in this direction, and it was well-received as evidenced by the following feedback.

> "The TAs are working fine, solving our queries and doubts. Nipun Sir is always there to help, and his way of explaining things is remarkable." - Anonymous feedback in Intro to Programming course
>
> "As a whole, the TA (Nipun Batra) is doing great in explaining the topics, and he makes the class interactive by relating things to the common world,·, having the attitude of teaching with relating to the common world and existing knowledge would help in a great manner!" - Anonymous feedback in Intro to Programming course
>
> "Relating to Real-World examples was the best part of the presentation." - Anonymous feedback in the PhD boot camp

While relating to the real world helps students see concepts based on everyday objects, I also believe that to develop a deep appreciation, it is often necessary to have a historical perspective. For example, the popularity of some of the text editors (like vim) and its relation to the network latency, can make the students appreciate the network latency timescale.

HIGH PERSONAL STANDARDS

As the adage goes, *well begun is half done*, I have always believed that the key role of an instructor is to inspire students. I have also maintained that one needs to set high personal standards as an instructor is often looked up to. To this effect, I took special care of a few things in my past teaching experience, like: being honest in admitting mistakes I make during lectures,

---

[1] https://relate.cs.illinois.edu/course/cs357-f15/file-version/03473f64afb954c74c02e8988f518de3eddf49a4/media/least-squares/Denoising.py based on my article https://nipunbatra.github.io/blog/2013/denoising.html

[2] https://nipunbatra.github.io/blog/2014/dtw.html

[3] https://nipunbatra.github.io/blog/2014/em.html

[4] https://nipunbatra.github.io/blog/2016/FT.html

and improve and correct them in subsequent lectures; being very punctual; putting in the required effort towards each lecture (once I casually did not put sufficient effort and it showed in my presentation); being impartial and non-judgemental, etc. The following testimonial showed that all that effort counts!

> "You were the first TA we had! Amazing experience, got me excited about academia and teaching" - Sukrit Kalra, BTech 2016, IIITD (in a personal conversation after getting PhD admits from Stanford, UC Berkeley, UW, CMU, Columbia)

PROJECT AND GROUP-BASED LEARNING

Similar to the above two points, I have a firm belief that students learn well when they can see something in action, something that they feel adds value, something which generates excitement, and something which they can show to others. I plan to introduce projects in different phases of the course and mandate the code to be released as open source. I also plan to encourage students to develop projects that can go live as new or replace existing institute systems.

INTUITIVE THINKING

I will try to pique the student interest in the subject area and allow them to see the intuitions behind things. I see this method being analogous to teaching a man to fish, instead of just feeding him food. I also try to create my problem sets to ensure that the students put their intuitive thinking cap on.

FEEDBACK-DRIVEN TEACHING

Throughout my teaching experiences, I have relied on implicit and explicit feedback to improve my future classes. I regularly took feedback from students to check if they understood the concepts clearly, or if they found themselves enough challenged to remain interested. I take student activity level and participation as implicit feedback during the lecture. I usually keep some interesting trivia/story/video as a backup in case I see that the class participation and interest is declining. As an example: while explaining code division multiple access (CDMA), playing a video showing the relative overlap/orthogonality between a pair of spoken languages (e.g., English <−> French) and how CDMA works best when the languages (codes) are orthogonal.

UG COURSES THAT I WOULD LIKE TO TEACH

1. Machine learning/Data Science/Data mining: I have worked significantly in these domains in my PhD and wrote several well-appreciated tutorials/articles.

2. Programming courses: I have taught one such course as a TA at IIITD. I love such courses as they allow me a chance to teach students who have freshly joined, and such courses allow me to leave a lasting interest in CS on the students.

3. Data structures and algorithms: While I have never taught such a course, I have an active interest in the subject area, and I continually keep brushing up my skills.

4. Computer Networking: Computer networking serves as a prerequisite for some of the research work that I have done.

PG COURSES THAT I WOULD LIKE TO TEACH

1. Seminar courses on one or more of the following topics: urban computing, computational sustainability, Cyber-Physical systems: I worked in these domains during my PhD.

2. Mobile computing/Internet of things: Both of these domains are related to my work and have similar basics.

3. Research methods: Not only have I taught this course during a PhD boot camp, but I have also led several discussions around labs in IIITD around the core ideas and practices of research methods. I am particularly interested in this course as it is an enabler for good research.

4. Recommender systems: I have worked on various aspects of recommender systems during my research.