Teaching Statement

Djellel Difallah (djellel@nyu.edu)

As a teacher, my goal is to train the next generations of successful engineers, scientists, and entrepreneurs. My aim is to prepare my students to keep up with the emerging field of data-driven science long after they finish their academic training. Thus, it is my responsibility to convey solid fundamentals and to spark a genuine interest in the subject. For this, I believe that it is important that the students develop a sense of curiosity, critical thinking and experimentation in order to sustain, and ultimately enjoy, the level of attention to details and patience necessary for success. In the following, I show how I implement this teaching philosophy with examples of my practices.

Problem Solving with Examples In order to spur students interest during the class, I consistently introduce multiple running examples and ask the students for input while elaborating the base cases. Next, I ask them to think about the corner cases and failure scenarios. Such thinking takes place in class or as a take home exercise and helps the students to fully digest new concepts. For example, in the Big Data course, after introducing the "Consistent Hashing" algorithm with some easy examples, I asked them to identify the limitations of the algorithm and to propose potential solutions. The discussion has led us to self-derive some actual improvement algorithms. I evaluate the efficacy of this method by observing the level of participation of the students as they need to fully grasp the concepts first in order to analyze the problem, propose solutions and/or ask questions. I also use online tools to distribute multiple choice questions that prompt the students for an input on their workstations during the class. These tools bring a novel dimension of interaction, I utilize them mainly as an icebreaker to prompt an answer from everyone in the class or to highlight the importance of a particular information.

Theory vs. Practice For an advanced graduate course, I interweave some research papers with the regular lectures. To help understand the selected scientific papers, I ask the students to read them ahead of the lecture and to write a one page summary of the contributions and key takeaways. The selection of the paper is key as I make sure that it depicts an algorithm (or a system) that they are going to experience during the labs. Reciprocally, the student will have a direct explanation of some tuning parameters of a system (e.g., size of partitions) introduced in a lab, and can relate to the experiments varying such parameters and their implication from the paper. I noted that the students were able to retain theoretical concepts that had practical implications which they experimented with. For example, by covering research papers on database column stores, in the final examination the students were clearly able to recommend a row versus column store for a given database workload and to argument their choice theoretically and practically. I have received positive feedback from the students who enjoyed reading scientific papers when presented this way.

Project Personalization Creating a computer system from scratch is challenging and requires a lot of patience during the hurdles of implementation and debugging. To support an exciting learning-by-doing paradigm in my courses, I include a semester long project during which the students have the flexibility to propose and develop their own custom system around the main themes of the course. For example, one of the projects I designed and supervised as a Teaching Assistant consisted of creating a social platform. The students worked in groups to come up with an idea, develop it and present a demo at the end of the semester in front of their classmates and external guests. While working with the students, I noticed that they were gradually utilizing and implementing concepts from the lectures and applying them to their system.

Teaching Interests

As a professor, I am prepared to deliver undergraduate and graduate course. For example, "Algorithms and Data Structures" would cover the building blocks for reasoning about the storage and processing of data in addition to introducing time and space complexity analysis tools. "Introduction/Advanced Database Systems" are undergraduate, respectively graduate level courses that would cover the fundamentals of database design, SQL, as well as system internals, a sample project would be to create a database management system from scratch. "Applied Data Science", a graduate course that would cover popular machine learning algorithms in class and ground them in real use cases during lab sessions and projects (e.g., using Kaggle competitions, or Crowdflower public datasets).

To conclude, my teaching strategies have roots in my research process as I setup the students to become genuinely interested in solving the problem at hand. This naturally encourages them to invest the necessary time to research further techniques to identify and ultimately improve their current solution.