Teaching philosophy statement

Dr. rer. nat. Martin Bies

In this document, I explain my teaching philosophy: My motivation, how I design a course and assignments, how I provide a consistent and transparent evaluation scheme and a comfortable environment for effective learning. I provide examples from my past, in particular the course *Math 313* in spring 2021 at the *University of Pennsylvania*.

Motivation

By upbringing, I am a physicist. In contrast to mathematics, physics can often not be formulated in a rigorous manner. When I was a student, this often felt as being taught that "every animal has four legs" and, once I discovered snakes, being told that "snakes are pathological examples". It is probably understandable that this experience can be both frustrating and unsatisfying. A key motivation for me is to do better and to teach the way that I would have loved to be taught. In particular, I try to make complete statements ("all dogs, which have not suffered a birth trauma or accident, have four legs") and make a clear distinction between assumptions (e.g. the least-upper-bound property) and irrefutable logical deductions. Based on my experience, this is very well perceived.

Course design

In the information age, knowledge can be overwhelming. To my mind, it is therefore even the more important to explain why we study a certain topic and why we do it in a particular manner. In a backward design, it can thus be beneficial to first formulate learning goals and then organize the course towards achieving these goals [1].

The goal of *Math 313* is to convey an understanding of linear algebra which is sufficient to understand applications and to steer the corresponding functionality in computer algebra systems. For many students in neighboring disciplines such as data sciences, computer sciences and engineering, *Math 313* is a mandatory course requirement. In those disciplines, the programming language of choice is often `Python` (e.g. *scikit-learn* [2] or *PyTorch* [3] in data science). Hence, it seems reasonable to make a selection of important linear algebra applications, exemplify each in `Python` and develop the relevant theory:

1. Goal: Solving linear equations.
   Theory: PLU-factorization, linear subspaces (e.g. the kernel), matrices as maps.

2. Goal: Linear regression and Taylor expansion.
   Theory: Orthogonality in vector spaces.

3. Goal: Tell how many solutions a linear system has.
   Theory: Determinants.

4. Goal: Markov processes, Page rank algorithm, solving systems of coupled ODEs.
   Theory: Eigenvalues, eigenvectors and the spectral theorem.

I introduced determinants by their properties and then proved existence, uniqueness and formulated rules for their computation. This approach gives a satisfactory answer to "what is a determinant" and exemplifies the important concept of defining a quantity by its properties. I completed the course with a brief exposition of signal processing via the singular value decomposition and the complex Fourier transform via complex matrices in favor of the classical climax of linear algebra courses – the Jordan normal form.

Assessment

Students want and deserve feedback – not only to learn from their mistakes, but also to gauge how they can get an A. Importantly, students also use metadata (organization of the course, context) to formulate solutions that they believe the instructors expects them to find. This is termed "lecture theory" and applies mutually to how students approach exams. As *Math 313* instructor, I formulated clear rules to fix this context:

- Weekly assignments – 30% of final grade,
- Two midterm exams – each 20% of the final grade,
- Final exam – 30% of the final grade.

In the first lecture, I informed the students that all results *proven* in the lecture can be used – a reference to "lemma x.y.z" is sufficient – and that any other results require justification. What I expect for the latter, I then presented twice-weekly in the lecture.

The assessment rules must also be properly communicated to the TAs/graders. Since I believe that positive reinforcement is a stronger motivation than punishment, I encouraged my *Math 313* grader to look for correct results and award points (rather than subtracting for mistakes). Also, I met my grader once weekly. Before this meeting, I send out a draft of the upcoming assignment and solutions. We then discussed possible issues (e.g. the formulation of the exercises), expectations for the solutions and which of the previous exercises the students found most challenging (gauge lectures/assignments).

Structure of lectures

I start my lectures with a brief recap of the previous lecture, then provide a skeletal structure of today's lecture and try to provide answers to "why" we study this very topic today and "why" we approach it in this way.

For the body/main content of the lecture, it is worth remembering that the attention span of a healthy person ranges from 10 to 20 minutes. To accommodate this, I try to take a small break after having formulated/proven a central result. This allows the students to absorb this new insight and formulate questions. I also use these breaks to actively encourage questions and, if not asked, ask critical questions myself to discuss the result. My impression is that these breaks help to refocus/collect and give me the opportunity to regain attention that I may have lost "in the depths of an involved proof".

As a personal touch, I use the last lecture to summarize the course. I believe that this summary helps the students to see the bigger picture and, at the same time, inspire and assist them in their preparation for the final exam.

Lecture materials

As a student, I found it very convenient if a course provided lecture notes. It establishes a common resource for the students and graders/TAs to discuss the course content, be informed on the progress of the course (e.g. if a student fell sick) and reference results ("Lemma x.y.z. in the notes implies ..."). For *Math 313*, I provide typeset notes on my website. Another important resource for the students are solutions to the weekly assignments, which I provided in handwritten form. During the virtual spring semester 2021, I could also easily record the lectures and make these recordings available to the students. Lastly, I was in touch with the library to provide scans from the book that this course follows, so that additional material was available for interested students.

Weekly exercises

I believe that own discoveries reach deeper than read knowledge. Hence, if possible, I have the students uncover results by themselves. For example, towards the beginning of *Math 313*, I gave the students a Markov matrix M and asked them to compute $M^n\vec{v}$ for an initial state $\vec{v}$. This example was inspired from a real-life problem with steady state. With `Python`, it is not too hard to approximate this steady state. In doing so, the students not only acquired proficiency with performing such elementary matrix computations in `Python`, but they also uncovered basic facts about Markov processes. In another exercise, I asked for the Taylor series of a step function. With `Python`, the students could compute and plot this to very high order. They were fascinated that even at order 1000 there were significant differences. To stimulate them further, I asked to repeat this task for a Gauss distribution, for which even at 2nd order almost no difference is visible.

Integration

Often, course evaluations are conducted towards the end of a semester. In this case, the feedback is received too late to improve the learning experience for current classes. When I TA'ed my very first course, I therefore designed my own poll and asked the students for anonymous feedback towards the middle of the semester. This was very helpful (e.g. I was asked at the time to provide a more detailed recap of the lectures) and is by now the standard at my alma mater. Along the same lines, I designed a *Canvas quiz* for *Math 313*. Since I had never studied in the US, the most challenging part for me was to choose the level of difficulty and the amount of course work correctly. Of course, I reached out to colleagues during the preparation of the course. Still, the feedback that I received from the students via my *quiz* was very helpful for further fine tuning.

In *spring 2021*, many students suffered hardships due to *COVID-19*. Of course, I tried to help the students under these circumstances, e.g. by granting generous extensions for the submission of homework. I offered a weekly office hour. For students in largely different time zones (e.g. Hong Kong), I provided flexible and individual meeting times after a brief email exchange. I also asked my grader to hold an office hour, where the students could ask grading related questions.

A personal opinion

I conclude with the quote that a student sends me by email towards the end of the spring semester 2021:

"Last semester I took Math 240, and it served as my first exposure to linear algebra. I did not even remotely enjoy that course, and had a distaste for linear algebra as I truly did not understand it or any of the value that learning it posed. After taking 313, I can tell you that I now intend to pursue a math minor to continue my mathematical education alongside my computer science/data science major. I feel that I have developed a very good understanding and appreciation for this subject matter, and I owe that entirely to your logic driven explanation oriented teaching style. You do a fantastic job of always addressing WHY things are the way they are, rather than just giving us formulas and examples, and it has made 313 one of my favorite Penn classes to date."

References

[1] Davila Rubio, Alejandro Mauricio, *Wiggins, G., & McTighe, J. (2005) Understanding by design (2nd ed.). Alexandria, VA: Association for Supervision and Curriculum Development ASCD*, *Colombian Applied Linguistics Journal* **19** (2017) 140, URL `http://dx.doi.org/10.14483/calj.v19n1.11490`. 1

[2] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., *Scikit-learn: Machine Learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830, URL `https://scikit-learn.org/stable/`. 1

[3] Paszke, Adam and Gross, Sam and Massa, Francisco and Lerer, Adam and Bradbury, James and Chanan, Gregory and Killeen, Trevor and Lin, Zeming and Gimelshein, Natalia and Antiga, Luca and Desmaison, Alban and Kopf, Andreas and Yang, Edward and DeVito, Zachary and Raison, Martin and Tejani, Alykhan and Chilamkurthy, Sasank and Steiner, Benoit and Fang, Lu and Bai, Junjie and Chintala, Soumith, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019, URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`. 1