PSY4218/6218 Computational Cognitive Modeling
Furman 209
Tu/Th 11:00-12:15
Spring 2020

Thomas Palmeri
507 Wilson Hall
343-7900 (office)
thomas.j.palmeri@vanderbilt.edu
office hours: Wed 2:00-3:00 or by appointment

Course Overview

This course provides an overview of the how-tos and the whys of computational cognitive modeling. This course is not intended to be a general survey of computational models of human cognition, although we will use several important models over the course of the semester. Instead, we will focus on what models are, why we use models, how to recognize good modeling versus bad modeling, how to implement a model, how to fit a model to data, how to evaluate the fit of a model, how to compare and contrast competing models, how to evaluate special cases of a model, and how to develop and test new models. We will talk about a number of practical issues involved in implementing models. We will primarily talk about models that account for response probabilities, response times, and neurophysiology in a few selected domains. We will talk about why we develop and test models, when it is appropriate and inappropriate to test models, what kinds of choices are made when developing a model, what are the best ways to use modeling most effectively, and what we can learn from models. Many of the techniques and issues we talk about apply to all kinds of modeling, from abstract macro-level cognitive models to micro-level neural models.

By taking this course, you will be able to implement models, simulate models, generate model predictions, fit models to data, and contrast competing models. This course will also give you the tools and background to take a more critical eye to modeling work you might read in the literature. We'll cover a variety of practical issues like using MATLAB or Python for modeling, random number generators, Monte Carlo simulations, using the high-performance computing facility at ACCRE, speeding up simulations, using bootstrapping techniques, and simulating differential equations.

Programming Languages

I will try to give examples in both Matlab and Python, but I may only have Matlab examples sometimes because the previous semesters I have taught this course used Matlab. I strongly encourage students to use Matlab or Python for homework assignments; I will also accept R if that is the language a student is most proficient in (I will not give examples or point to libraries or talk about the nuances of R because it is not a programming language I use). If you have never used Matlab, Python, (or R), you will need to learn how to program in one of those languages while learning the course material. I cannot accept homework assignment in Java, C++, or any other programming language. If you have never programmed before, this is not the course for you.

With Matlab, you will need to have the Optimization, Global Optimization, Statistics, Image Processing, Signal Processing, and Parallel Computing toolboxes installed. With Python, you will want to install the most recent Anaconda build (Python 3.7), which includes all of the modules you will need for the course. With R, you are on your own to figure out what you need to complete the course assignments.

Course Requirements

Class Participation (5%) **:** I want people to show up, discuss, disagree, and ask questions. If you're going to miss class, please let me know beforehand. Please do the readings for the week before class.

Homework Assignments (70%) **:** There will be a homework assignment most weeks. The assignments will usually involve implementing something in MATLAB or Python. I may give some the skeleton code ahead of time, so the programming should not be overly excessive. I have no problem with people talking together about how to do the assignments conceptually and helping each other out with programming issues, but everyone needs to do their own assignment.

Final Project (25%) **:** The final project gives you the opportunity to do something related to the course that is also relevant to your own research. The most obvious thing to do would be to implement a model in MATLAB or Python (or R) and test its predictions or fit it to some data. You could also develop and test a new model. Or you can systematically compare different ways of simulating a model or fitting a model to data. We will talk more about the project a bit later in the semester. You might start to think about what modeling you might want to do for the project. Feel free to talk with me. Talk to your advisor. I want you to do something that will be useful for your own research. We'll set aside time at the end of the semester for some short presentations of what you did (required for graduate students, extra credit for undergraduates). I want you to turn in your running code (that I can run myself), the modeling results (figures, graphs, and tables), and a brief description of what you did (a few pages would be fine). The goal is to implement/test a model using the techniques we have discussed in class, to do something with a model, not to write a paper about the model. Projects are due the last day of class.

Course Readings

The primary text for this course is: Farrell, S., & Lewandowsky, S. (2018). *Computational Modeling of Cognition and Behavior*. Cambridge University Press.

Other readings will be distributed on Brightspace. I will also add supplemental readings to Brightspace.

Brightspace

Class topics, assigned readings, supplemental readings, and homework assignments will be posted on Brightspace. Homework assignments will be turned in on Brightspace and grades and comments will appear on Brightspace.