Model AI Assignments 2019

Todd W. Neller[1], **Raja Sooriamurthi**[2], **Michael Guerzhoy**[3], **Lisa Zhang**[4], **Paul Talaga**[5],
Christopher Archibald[6], **Adam Summerville**[7], **Joseph Osborn**[8], **Cinjon Resnick**[9], **Avital Oliver**[10],
Surya Bhupatiraju[11], **Kumar Krishna Agrawal**[10], **Nate Derbinsky**[12], **Elena Strange**[12],
Marion Neumann[13], **Jonathan Chen**[13], **Zac Christensen**[13], **Michael Wollowski**[14],
and Oscar Youngquist[14]

[1]Gettysburg College, [2]Carnegie Mellon University, [3]Princeton University, [4]University of Toronto Mississauga,
[5]University of Indianapolis, [6]Mississippi State University, [7]California State Polytechnic University, Pomona,
[8]Pomona College, [9]New York University, [10]Google Brain, [11]DeepMind Technologies, [12]Northeastern University,
[13]Washington University in St. Louis, [14]Rose-Hulman Institute of Technology

Abstract

The Model AI Assignments session seeks to gather and disseminate the best assignment designs of the Artificial Intelligence (AI) Education community. Recognizing that assignments form the core of student learning experience, we here present abstracts of ten AI assignments from the 2019 session that are easily adoptable, playfully engaging, and flexible for a variety of instructor needs. Assignment specifications and supporting resources may be found at http://modelai.gettysburg.edu.

Implementing a Recommender System using MapReduce - Raja Sooriamurthi

This assignment combines two prominent machine learning and big data technologies: MapReduce and recommendation. Nearly 20 years ago, as Google scaled, they created several new technologies. Foundational amongst those technologies are the resilient Google File System (GFS) and a computing paradigm known as MapReduce. Google published their work in a few highly influential papers. Researchers, inspired by these descriptions, created open source versions, which eventually became the big data platform now known as Hadoop. Yelp created a Python based library to write MapReduce programs known as mrjob. Based on user preferences, recommender systems produce an ordered set of recommendations. In this series of exercises, students gain hands on experience with how user-based, item-based, and content-based recommender systems work. Modeling the core computation in a spreadsheet helps convey the essence of these algorithms. With this background students then express the recommender algorithm in the MapReduce paradigm using mrjob. Experiments are done using the Movielens data set[1]. The assignment also forms the context for discussing the Netflix prize.

Building a Fake News Detector - Michael Guerzhoy and Lisa Zhang

In this assignment, students build a fake news detector. We supply a dataset with news headlines labelled as "fake" or "real", and students build several models for classifying the headlines: a Naive Bayes classifier, a logistic regression classifier, and a decision tree. Students improve on the fake news detector they built by using more sophisticated algorithms and by collecting their own training sets.

We emphasize model interpretation: students analyze their Naive Bayes and logistic regression models to obtain the words whose presence and absence indicate "fake" and "real" most strongly. Students visualize the decision tree that they build, and generate new data from the Naive Bayes model.

We hold a Kaggle-style class-wide competition, and award bonus marks to student teams that classify our secret hand-curated fake news dataset most accurately. Participating in the competition allows students to experiment with state-of-the-art text classification algorithms, follow best practices (such as using ensembles of classifiers), and be exposed to the issues involved in assembling a training set without knowing what the test set will look like.

Using Ultimate Tic Tac Toe to Motivate AI Game Agents - Paul Talaga

Ultimate Tic Tac Toe is an extension to the classic Tic Tac Toe game. In an introductory AI course covering adversarial search methods, Tic Tac Toe is a good game to demonstrate decision-making algorithms by way of its simple rules, gameplay, and relatively small state space. In this assignment, students are tasked with extending the Tic Tac Toe game code to Ultimate Tic Tac Toe, adding an AI agent, and/or experimenting with parallel implementations. Or, students could be provided with a working Ultimate Tic Tac Toe implementation and asked to add an AI agent. If a common gameplay program and game state structure is mandated, AI agents could be pitted against each other in a tournament or for grading purposes.

[1]https://grouplens.org/datasets/movielens/

RISK AI Project - Christopher Archibald

This project requires students to design and implement agents for the classic board game of RISK. The main assignment consists of an open-ended final project where students can use topics covered in introductory AI courses to create their agents, including path-planning, game-tree search, reinforcement learning, and supervised learning. Two preliminary RISK assignments, which focus on path-planning with uniform cost search and supervised learning with decision trees, are also included. The game of RISK involves players competing to take over all of the territories on a world map, with battles being decided stochastically by rolling dice. Successful agents must make many different kinds of complex decisions. The project materials include a complete RISK simulator and graphical game viewer which allows game logs to be viewed. The game of RISK is unique in that it can be played with anywhere from two to eight players, and the project framework includes support for multiple tournament formats for evaluating the student agents. Several baseline agents are provided for students testing. This project gives students the opportunity to find creative ways to use ideas from an AI course in an engaging and well-known game that makes for a fun and rewarding culminating project.

The Minecraft Projects - Adam Summerville and Joseph Osborn

DeNero and Klein's Pac-Man Projects (DeNero and Klein 2010)[2] are a landmark suite of assignments covering a broad variety of AI techniques. Driven by similar concerns, we use the game Minecraft as a motivating context for AI assignments. Students will likely be familiar with Minecraft (as of 2016, nearly 107 million copies had been sold), and it is becoming better-known in the AI community due to Microsoft's Project Malmo platform and the 2018 AIIDE MARLÖ Workshop. One unusual aspect of these assignments is that exceptional homework could conceivably translate into novel research publications!

We have developed four assignments covering key areas: path planning, task planning via iterative widening, and scheduling and production planning by linear (integer) programming. Each can be scaled or scaffolded more or less, as the class requires, and each features an automatic grader and reflection questions for students' analytic work. While these exercises use simplified Malmo-like environments, the agents' actions could readily be realized in Minecraft worlds using the Malmo API. We provide Python starter code; Malmo's API also supports Java, C#, and other languages via an OpenAI Gym API. A Conda environment and setup instructions for working the exercises is also given.

Depth First Learning: DeepStack - Cinjon Resnick, Avital Oliver, Surya Bhupatiraju, and Kumar Krishna Agrawal

We present a Model AI assignment targeting the DeepStack paper. It consists of six sessions, each of which roughly correspond to 6-10 hours of work (a 1.5 hour class, readings, and assignments). The sessions have both required and optional readings, as well as problems where students must demonstrate their understanding by either proving a mathematical statement, answering a question, or implementing code. The assignment is modeled after the Depth First Learning[3] approach. There, students first build a base of understanding by developing competency in the dependency tree of concepts underlying the paper. Only after that do they read the paper. Over the first five sessions, the students dive into game theory and counterfactual regret minimization. By the time they read the paper in the sixth week, their thorough grasp of the background material makes it easy to fully understand the paper's results, its significance, and which research directions are left under-explored.

Nearest Neighbor Classification (with almost no background) - Nate Derbinsky and Elena Strange

Humans get better with experience and, using Machine Learning, computers can too âĂŞ– and with this assignment, students with little-or-no programming experience can gain conceptual and hands-on exposure to Nearest Neighbor classification. The included variants are (1) a no-experience version, for students participating in a one-time workshop; and (2) an early-semester CS1 version, for students who have had some introduction to programming with Python. The supplied materials include a homework assignment (in Python, with practice problems, rubrics, starter files, auto-grading unit tests, and an explanatory handout) as well as a presentation with type-along code for CS/AI outreach events. This project is optimized for minimizing prerequisite knowledge, and so leaves a universe of expansion possibilities, including language choice, dataset size/domain, and inclusion of advanced concepts (e.g., non-linear search algorithms, kernel representations).

Introduction to Python for Data Science - Marion Neumann and Jonathan Chen

We provide an interactive guided lab to introduce Python for data science (DS). We provide two Jupyter notebooks introducing the basics of Python and the DS workflow using the Iris dataset. We interactively introduce expressions, variables, strings, printing, lists, dictionaries, control flow, and functions to students that are already familiar with a programming language from an introductory CS course. The second lab aims at motivating students to acquire skills such as using statistics to model and analyze data, knowing how to design and use algorithms to store, process, and visualize data, while not forgetting the importance of domain expertise. We begin by establishing the example problem to be studied based on the Iris dataset. The next step is to acquire and process the data, where students practice how to load data and how to process strings into numeric arrays. Then, we explain different plotting methods such as box plots, histograms, and scatter plots for data exploration. Finally, we

split the data into training and test set, build a model, use it for predictions, and evaluate the results. The main learning objectives are to get to know and practice Python in the context of data science and machine learning.

Introducing the Data Science Workflow using Sentiment Analysis - Marion Neumann and Zac Christensen

We provide an interactive guided lab with a follow-up homework assignment to introduce the basic data science workflow by exploring sentiment analysis. The lab focuses on introducing the machinery using a given dataset of movie reviews and the assignment highlights data acquisition and exploration. After introducing sentiment analysis, we explain a simple rule-based approach to predict the sentiment of textual reviews using three handcrafted examples. This introduction shows how to pre-process text data and how to use lists of positive and negative expressions to compute a sentiment score. Then students will implement the approach to predict the sentiment of movie reviews and evaluate the results. The lab concludes with a discussion of the limitations of the rule-based approach and a quick introduction to sentiment classification via machine learning. The homework assignment reiterates over the process of building and analyzing a sentiment predictor with the focus on collecting and preprocessing their own dataset scraped from twitter using an API. The main learning objective of this activity is getting to know the inference problem and walking through the entire data science workflow to tackle it. This module only requires minimal programming background and is an ideal precursor to introducing machine learning.

A Gentle Introduction to the Backpropagation Algorithm and Feedforward Networks - Michael Wollowski and Oscar Youngquist

This assignment is designed for students to learn about the algorithmic underpinnings of feedforward networks by implementing and tuning various perceptron and feed-forward neural networks. Students are asked to implement a basic perceptron and to modify given code for a feedforward network so as to grasp the basic workings of neural networks. Students are furthermore asked to tune various hyperparameters for a given feed forward network so as to learn about the effect of those parameters on effectively and efficiently training various functions that are typically introduced when neural networks are first introduced in class. These functions include the Boolean AND and XOR functions. To reinforce this understanding and enable students to develop an appreciation of the importance of data curation, they are then asked to implement a digit recognition network, using the MNIST data set, using the given feedforward network.

References

DeNero, J., and Klein, D. 2010. Teaching introductory artificial intelligence with pac-man. In *Proceedings of the 2010 AAAI Symposium on Educational Advances in Artificial Intelligence*. https://www.aaai.org/ocs/index.php/EAAI/EAAI10/paper/view/1954.