ESE 2040: Decision Models

Instructor: Nikolai Matni, Assistant Professor, Dept. of ESE,
TAs: Fengjun Yang, Thomas Zhang, Alex Nguyen-Le
Graders: Samar Hadou
Course website: https://nikolaimatni.github.io/courses/ese2040-fall2022/
Lectures: Tu/Th 1:45pm-3:15pm ET, ANNS 111
Recitations: Fr 1:45pm-3:15pm, ANNS 111. In each, a TA will cover the solutions to some of the problems in the study packet that will be relevant for that week's homework, as well as relevant programming concepts. Any remaining time can be used for Q&A.
Other resources: We will use CANVAS to disseminate course materials, and Ed Discussion for class related discussions.

Office hours

All teaching staff (TAs, instructor) will hold 1 hour of office hours each week. Times and locations will be posted on CANVAS. If you plan to attend, *please let the relevant person know in advance*, so that we can manage congestion.

Course description

This first course in decision making will introduce you to quantitative models for decision and design in the sciences, engineering, machine learning, data science, logistics, and economics. Through application-based case studies, you will be shown how to (i) formalize a decision problem as a mathematical optimization problem, and (ii) solve the resulting optimization problem using Python scientific computing modules. You will also be given a brief introduction to the optimization algorithms and programming tools underpinning contemporary deep learning and shown how to apply them to decision and design problems.

Prerequisites

The only official prerequisites for this class are Math 1400 and Math 1410. Basic familiarity with Python and programming is helpful, but not necessary.

Course Materials

There is no official textbook or proprietary software needed for this class. We will share lecture notes, assignments, and slides on CANVAS. All programming exercises will be performed using Python Notebooks. We ask you to please not share course materials with those not registered in the class.

In addition, the following freely available textbooks may be useful as references:

- *Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares* by Boyd and Vandenberghe, available online.
- *Convex Optimization* by Boyd and Vandenberghe, available online.

Grading

- **Homework (50%)**: there will be 10 homework assignments. They will be assigned weekly, handed out on Tuesday at 1pm and due the following Tuesday at 1pm. There will be suitable breaks in assignments to accommodate the midterm and the Thanksgiving holiday weekend. Assignments will include both conceptual (written) and implementation (programming) exercises. You will be given ***5 free late days*** which you may use as you please throughout the semester, after which no late assignments will be accepted. Each homework problem will be graded on a scale of 0-2: no points are awarded for a skipped problem, 1 point for a solid attempt, and 2 points for a mostly correct solution.
- **Midterm exam (25%):** the midterm will consist of an in-class written component (15%) and a take-home computational component (10%). The in-class component will be closed-book and closed-notes. However, you will be allowed a single sheet of standard-sized paper with you with anything you want written on it (double-sided). No electronic devices are allowed. The take-home component of the exam will be open book.
- **Final exam (25%):** there will be an in-class final exam scheduled during the final exam period with time and date set by the Registrar. It will be closed-book and closed-notes. However, you will be allowed a single sheet of standard-sized paper with you with anything you want written on it (double-sided). No electronic devices are allowed.

Collaboration Policy

- **Homework:** You are allowed and in fact encouraged to **discuss homework problems with your peers,** but you must **conceptualize and write up** your own homework solutions and code to hand in. If you discuss a problem with someone for more than five minutes, you are required to list their name on the first page of your assignment as a collaborator. Full marks will still be awarded to all collaborators.
- **Midterm and Final exam:** no collaboration or discussion with classmates is allowed.
- **Code of Academic Integrity:** All students are expected to adhere to the University's Code of Academic Integrity.

Course Outline

Please note that this is a ***tentative*** outline, and that the topics, their components, and their order may be adjusted or removed during the semester.

<u>Part I: Deterministic modeling and optimization</u>

- **Systems of linear equations**
  - *Motivating applications:* Balancing chemical reactions, Diffusion systems (e.g., current, transportation), Leontif economy input-output models
  - *Technical concepts:* Review of Math 1410 concepts—over and under-determined systems of linear equations, matrix-vector notation, linear-independence and matrix inverse
  - *Computational tools:* python, ipynb notebooks (colab), numpy, 1d, 2d, and n-d arrays, numpy.inv, numpy.linalg.solve, matplotlib
  - *Estimated time to complete:* 1 week (2 lectures + 1 recitation + 1 hwk)

- **Least squares**
  - *Motivating applications:* Advertising purchases, Equalizer design, Network Tomography, K-means clustering on MNIST
  - *Technical concepts:* Overdetermined systems with/without solutions via range of tall matrix A, norm as notion of distance/size and euclidean norm, column/row interpretation of least-squares problem, level curves, solution via calculus and normal equations, (optional: orthogonal vectors and orthogonality principle, matrix least-squares), K-means algorithm via iterative least-squares
  - *Computational tools:* numpy.pinv, numpy.linalg.lstsq, sklearn.datasets, sklearn.cluster.kmeans, intro to object oriented programming
  - *Estimated time to complete:* 1 week (2 lectures + 1 recitation + 1 hwk)

- **Data-fitting with least squares**
  - *Motivating applications:* time-series data, estimation of trend and seasonality, house price regression, temperature prediction
  - *Technical concepts:* data (feature, label), linearly parameterized models, least-squares model fitting via regression, linear and polynomial fit, AR models, validation, model-selection and feature engineering, cross-validation.
  - *Computational tools:* sklearn.model_selection.cross_val_and_score
  - *Estimated time to complete:* 1 weeks (2 lectures + 1 recitation + 1 hwk)

- **Linear programming**
  - *Motivating applications:* Diet problem and profit maximization, Markowitz portfolio optimization, Transportation and network flow problems: single and multi-commodity flow, Assignment problem with application to cryptocurrency arbitrage, Game theory (rock/paper/scissors)
  - *Technical concepts:* linear inequalities, LP formulation, matrix/vector formulation, graphing the feasible set, solutions = vertices, Duality and sensitivity analysis

- *Computational tools:* cvxpy for solving linear programming problems
- *Estimated time to complete:* 2 weeks (4 lectures + 2 recitations + 2 hwk)

- **Constrained & multi-objective least-squares**
  - *Motivating applications:* gene selection, identifying key factors from marketing data, spam detection, image deblurring and inpainting, least-squares control, denoising, and estimation
  - *Technical concepts:* multi-objective LS + regularization, effects of different regularizers, Ridge Regression and LASSO, quadratic and TV smoothing, regularization path and pareto-optimal surface, (optional: cross-validation for weight selection)
  - *Computational tools:* cvxpy for constrained and regularized least-squares
  - *Estimated time to complete:* 1 weeks (2 lectures + 1 recitations + 1 hwk)

- **(Mixed) Integer Programming**
  - *Motivating applications:* Traveling salesman problem, Warehouse placement problem, Airline scheduling, Fault detection, Sudoku
  - *Technical concepts:* boolean and integer problems, pure IP, MILP, (optional: MIQP), encoding logical constraints as IPs
  - *Computational tools:* cvxpy for MIPs
  - *Estimated time to complete:* 1 weeks (2 lectures + 1 recitations + 1 hwk)

- **Autodiff and Gradient descent**
  - *Motivating applications:* image classification, robotic nonlinear control/RL, biomedical imaging
  - *Technical concepts:* Nonlinear least squares, Gradient descent basics, backtracking and line-search for step size selection in GD, autodiff as a blackbox, GPUs and vectorization
  - *Computational tools:* intro to jax, jax.numpy, jax.jacobian, jax.jvp, vectorization via jax.vmap, just-in-time-compilation via jax.jit.
  - *Estimated time to complete:* 1 weeks (2 lectures + 1 recitations + 1 hwk)

<u>Part II: Probabilistic modeling and optimization</u>

- **Risk/Moment optimization**
  - *Motivating applications:* diet and profit problems with uncertain costs, minimum risk portfolio optimization, (optional: chebyshev inequalities + election polling)
  - *Technical concepts:* probability review: probabilities, expectation, variance, discrete and continuous r.v.s, measures of risk and risk regularized optimization
  - *Computational tools:* numpy.random, jax.random, cvxpy
  - *Estimated time to complete:* 2 weeks (4 lectures + 2 recitation + 2 hwk)

- **Maximum likelihood estimation**
  - *Motivating applications:* Linear measurement model with stochastic noise (revisit least squares, ridge regression, and LASSO examples) with Gaussian noise ⇔ least-squares, Laplacian ⇔ LASSO, multi-class classification (MNIST, CIFAR-10, Imagenet)
  - *Technical concepts:* conditional probability, Bayes updating, and maximum likelihood estimation, (binary) cross-entropy loss derivation, feature engineering
  - *Computational tools:* nothing new this week
  - *Estimated time to complete:* 1 week (2 lectures + 1 recitation + 1 hwk)

- **A shallow introduction to deep learning**
  - *Motivating applications:* classification, NLP, deep RL, computer vision, robotics, etc., focus will be training an MLP for MNIST and CIFAR-10 classification: learn the features from data!
  - *Technical concepts:* What is a neural network? Motivation for SGD and minibatches (scalability, vectorization), (optional: informal intuition mapping gradient estimator variance to SNR); if time allows: definitions of equivariance and invariance, convolutions for translational equivariance, pooling (max, avg) for invariance, convnets.
  - *Computational tools:* flax neural network package, optax
  - *Estimated time to complete:* 1 weeks (2 lectures + 1 recitation + 1 hwk)

Tentative homework and midterm exam schedule

As above, these dates are not fixed, and may be adjusted to the pace of the class.

- 09/06: Hw1 out
- 09/13: Hw2 out, Hw1 due
- 09/20: Hw3 out, Hw2 due
- 09/27: Hw4 out, Hw3 due
- 10/04: Hw4 due, midterm study break
- 10/13: In-class midterm, take-home midterm out
- 10/18: Hw5 out, take-home midterm due
- 10/25: Hw6 out, Hw5 due
- 11/01: Hw7 out, Hw6 due
- 11/08: Hw8 out, Hw7 due
- 11/15: Hw9 out, Hw8 due
- 11/22: Hw9 due, Thanksgiving break
- 11/29: Hw10 out
- 12/06: Hw10 due