EE266 and MS&E251: Introduction

About the course

Optimization

Dynamical systems

Stochastic control
About the course
About the course

- EE266 is the same as MS&E251
- Formerly called EE365
- Created by Stephen Boyd, Sanjay Lall, and Ben Van Roy in 2012
- Taught by Sanjay Lall this year
Control

- *multi-step decision making, in an uncertain dynamic environment*
- observe, act, observe, act, ...
  - your current action affects the future
  - there is uncertainty in what the effect of your action will be
- goal is to find *policy*
  - (computational) map from what you know to what you do
- called *recourse* or *feedback*, a richer concept than optimization
Applications

- multi-period investment
- automatic control
- supply chain optimization
- internet ad display
- revenue management
- operation of a smart grid
- data center operation

...and many, many others. What is the common abstraction?
Approach

- how to formulate and solve problems
- solution is usually an algorithm
- focus on ideas, not technicalities of corner cases
- similar style to ee263
- practical homeworks with extensive coding
Dynamics

intellectual components

- observe: statistical inference
- decide: optimization
- repeat: dynamics, with uncertainty

this course focuses on the consequences of dynamics, specifically:

- dynamic programming
- for Markov decision processes
Prerequisites

▶ linear algebra (EE263 or MS&E211; more than Math 51)

▶ probability (EE178/278A or MS&E220)

▶ not dependencies, but may increase appreciation:
  ▶ other classes in control
  ▶ artificial intelligence, Markov chains, optimization
Curriculum

- MS&E251 in the MS core, and in *decision and risk analysis*

- EE&266 satisfies MS breadth, and in two depth sequences:
  - *control and system engineering*
  - *dynamical systems and optimization*
Administration

- the website ee266.stanford.edu
- piazza, coursework
- 70% final, 30% homework
- 24-hour take-home final exam
Books

- Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming (online)
- Bertsekas, Dynamic Programming and Optimal Control, vol. 1
Optimization
Optimization problem

\[
\text{minimize} \quad f(x) \\
\text{subject to} \quad x \in X
\]

- \(x\) is decision variable (discrete, continuous)
- \(X\) is constraint set
- \(f : X \rightarrow \mathbb{R}\) is objective (cost function)
- \(x\) is feasible if \(x \in X\)
- \(x\) is optimal (or a solution) if \(f(x) = \inf_{z \in X} f(z)\)
- \(f\) and \(X\) can depend on parameters (data)
- can maximize by minimizing \(-f\) (reward, utility, profit, . . .)
- standard trick: allow \(f(x) = \infty\) (to embed further constraints in objective)
Solving optimization problems

- a solution method or algorithm computes a solution, given parameters
- difficulty of solving optimization problem depends on
  - mathematical properties of $f, \mathcal{X}$
  - problem size (e.g., dimension of $x$ when $x \in \mathbb{R}^n$)
- a few problems can be solved ‘analytically’
- but this is not particularly relevant, since we adopt algorithmic approach
Examples

- find shortest path on weighted graph from node $S$ to node $T$
  - $x$ is path
  - $f(x)$ is weighted path length (sum of weights on edges)
  - $\mathcal{X}$ is set of paths from $S$ to $T$

- allocate a total resource $B$ among $n$ entities to maximize total profit
  - $x \in \mathbb{R}^n$ gives allocation
  - (maximize) objective $f(x) = \sum_{i=1}^{n} P_i(x_i)$
  - $P_i(x_i)$ is profit of entity $i$ given resource amount $x_i$
  - $\mathcal{X} = \{x \mid x \geq 0, \ 1^T x = B\}$
Dynamical systems
(Deterministic) dynamical systems

\[ x_{t+1} = f_t(x_t, u_t), \quad t = 0, 1, \ldots \]

- \( t \) is time (epoch, stage, period)
- \( x_t \in X_t \) is state
- initial state \( x_0 \) is known or given
- \( u_t \in U_t \) is input (action, decision, choice, control)
- \( f_t : X_t \times U_t \to X_{t+1} \) is state transition function
- called time-invariant if \( f_t, X_t, U_t \) don’t depend on \( t \)
- variation: \( U_t \) can depend on \( x_t \)
Idea of state

- current action affects future states, but not current or past states
- current state depends on past actions
- state is link between past and future
  - if you know state $x_t$ and actions $u_t, \ldots, u_{s-1}$, you know $x_s$
  - $u_0, \ldots, u_{t-1}$ not relevant
- state is sufficient statistic (summary) for past
Examples (with finite state and input spaces)

discrete dynamical system:

- $\mathcal{X} = \{1, \ldots, n\}$, $\mathcal{U} = \{1, \ldots, m\}

- $f_t : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ called transition map, given by table (say)

moving on directed graph $(\mathcal{V}, \mathcal{E})$:

- $\mathcal{X} = \mathcal{V}$, $\mathcal{U}(x_t)$ is set of out-going edges from $x_t$

- $f_t(x_t, u_t) = v$, where $u_t = (x_t, v)$
Examples (with infinite state and input spaces)

linear dynamical system:

- $\mathcal{X} = \mathbb{R}^n$, $\mathcal{U} = \mathbb{R}^m$

- $x_{t+1} = f_t(x_t, u_t) = A_t x_t + B_t u_t + c_t$

very special form for dynamics, but arises in many applications
Dynamic optimization (deterministic optimal control)

minimize \[ J = \sum_{t=0}^{T-1} g_t(x_t, u_t) + g_T(x_T) \]
subject to \[ x_{t+1} = f_t(x_t, u_t), \quad t = 0, \ldots, T - 1 \]

- initial state \( x_0 \) is given
- \( g_t : X_t \times U_t \rightarrow \mathbb{R} \cup \{\infty\} \) is stage cost function
- \( g_T : X_T \rightarrow \mathbb{R} \cup \{\infty\} \) is terminal cost function
- variables are \( x_1, \ldots, x_T, u_0, \ldots, u_{T-1} \)
  (or just \( u_0, \ldots, u_{T-1} \), since these determine \( x_1, \ldots, x_T \))
- just an optimization problem (possibly big)
- also called classical or open-loop control
Deterministic optimal control

- addresses dynamic effect of actions across time
- no uncertainty or randomness in model
- is widely used (often, by simply ignoring uncertainty in the application)
Stochastic control
Stochastic dynamical systems

\[ x_{t+1} = f_t(x_t, u_t, w_t), \quad t = 0, 1, \ldots \]

- \( w_t \) are random variables (usually assumed independent for \( t \neq s \))
- state transitions are nondeterministic, uncertain
- choice of input \( u_t \) determines \textit{distribution} of \( x_{t+1} \)
- initial state \( x_0 \) is random variable (usually assumed independent of \( w_0, w_1, \ldots \))
Objective

▶ objective (to be minimized) is

\[ J = \mathbb{E}\left( \sum_{t=0}^{T-1} g_t(x_t, u_t, w_t) + g_T(x_T, w_T) \right) \]

▶ \( g_t : \mathcal{X}_t \times \mathcal{U}_t \times \mathcal{W}_t \rightarrow \mathbb{R} \cup \{\infty\} \) is stage cost function

▶ \( g_T : \mathcal{X}_T \times \mathcal{W}_T \rightarrow \mathbb{R} \cup \{\infty\} \) is terminal cost function

▶ often \( g_t, g_T \) don’t depend on \( w_t \), i.e., stage and terminal costs are deterministic

▶ infinite values of \( g_t \) encode constraints

▶ objective is mean total stage cost plus terminal cost
Information pattern constraints

- information pattern constraint: $u_t \text{ depends on what you know at time } t$
  
  $$u_t = \phi_t(Z_t)$$

- $Z_t$ is what you know at time $t$

- $(\phi_0, \ldots, \phi_{T-1})$ is called policy

- goal is to find policy that minimizes $J$, subject to dynamics
Information patterns

- full knowledge (prescient): \( Z_t = (w_0, \ldots, w_{T-1}) \)
  - for each realization, reduces to deterministic optimal control problem
- no knowledge: \( Z_t = \emptyset \)
  - reduces to an optimization problem; called open-loop
- in between: \( Z_t = x_t \) (called state feedback)
  - a little more: \( Z_t = (x_t, w_t) \)

these are very different problems!
Example: Stochastic shortest path

- move from node $S$ to node $T$ in directed weighted graph
- minimize expected total weight along path
- edge weights are random variables, independent in each time period

Information patterns:

- no knowledge: commit to path beforehand (knowing distributions of weights, but not actual values)
- full knowledge: weights on all edges at all times are revealed before path is chosen
- local knowledge: at each node, at each time, weights of out-going edges are revealed before next edge on path is chosen
Example: Optimal disposition of stock

- sell a total amount $S$ of a stock in $T$ periods
- price (and transaction cost) varies randomly
- maximize expected revenue

Information patterns:

- no knowledge: commit to sales amounts beforehand
- in each time period, the price and transaction cost is known before amount sold is chosen