ECE 6950 - POMDPs

Prof. Vikram Krishnamurthy
Dept. Electrical & Computer Engineering
Cornell University
email vikramk@cornell.edu

This document contain all the slides that will be used during the class.
They can be downloaded from
`vikram.ece.cornell.edu/teaching`

Logistics

- Two 70 min classes per week. Tue/Thursday 10:10 am broadcast to Rhodes 312.
- 4 assignments. (important to do them)

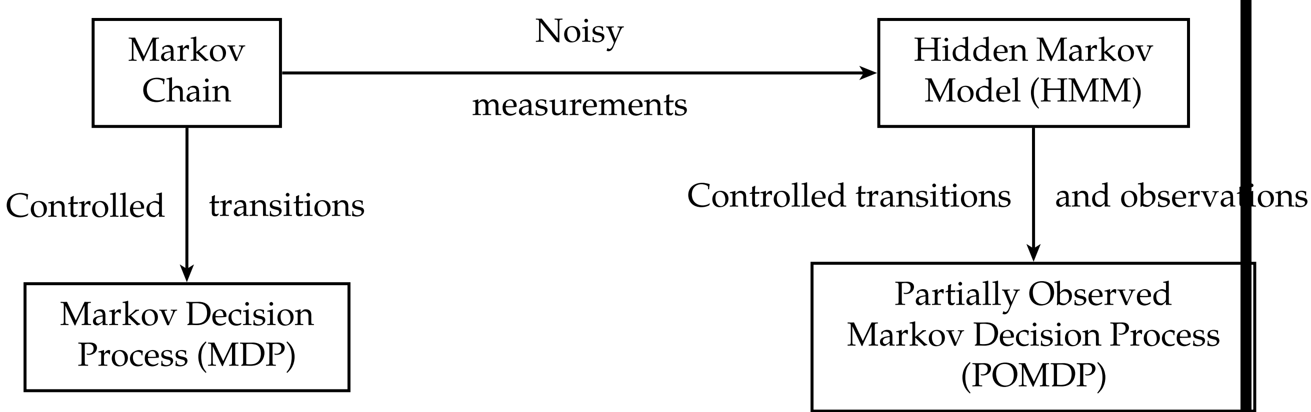
Assessment:

- 15% Class Participation (Ability to answer questions correctly in class).
- 40 % (4 Assignments)
- 45% (Take home final Exam)

Aim: To teach you sufficiently powerful analytical tools in Bayesian estimation and stochastic control (discrete time). We will occasionally use measure theoretic probability (but not required).

Textbook: The classes will follow closely the chapters in my book. *Partially Observed Markov Decision Processes - from Filtering to Controlled Sensing*, Cambridge Univ Press, 2016.

Big Picture



You will learn in this course about: Markov chains and their analysis (finite state), stochastic simulation, HMMs, Bayesian filtering, ML estimation, MDPs, POMDPs, structural results, reinforcement learning algorithms and their convergence analysis

OUTLINE

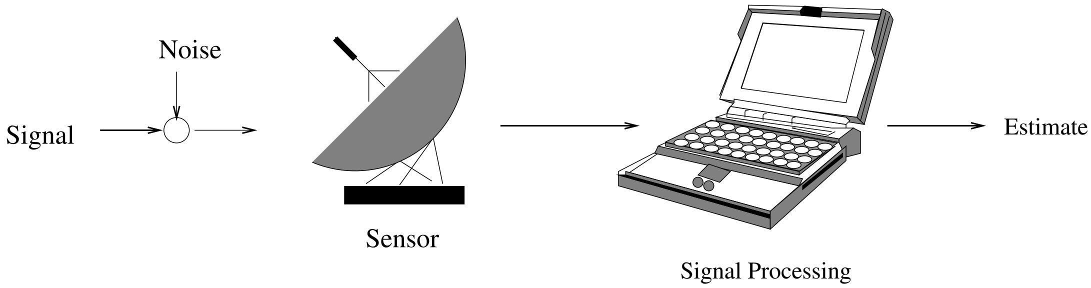
1. **Stochastic State Space models & Simulation** [5 hours]
   - Stochastic Dynamic Systems
   - Hidden Markov Models, Perron Frobenius Theorem, Geometric ergodicity
   - Linear Gaussian Models
   - Jump Markov Linear Systems and Target Tracking
   - Stochastic Simulation: Acceptance Rejection, Composition method. MCMC, Simulation-based optimal predictors
2. **Bayesian State Estimation**[5 hours]
   - Review of Regression Analysis and RLS.
   - The Stochastic Filtering Problem
   - Hidden Markov Model Filter
   - Kalman Filter
   - Particle Filters and Sequential MCMC
   - Filtering with non-standard information patterns: Non-universal filters, Social learning

3. **ML parameter Estimation** [3 hours]
   - Properties of ML estimators
   - EM Algorithm
   - Case study: Hidden Markov Models.
4. **Stochastic Gradient Algorithms & Stochastic Optimization**[4 hours]
   - Simulation-based Stochastic Optimization
   - Examples of Stochastic Gradient Algorithms
   - Ordinary Differential Equation analysis
   - Discrete Stochastic Optimization & Bandits
5. **Discounted Cost Markov Decision Processes: Full Observed case** [3 hours]
   - MDP models and examples
   - Stochastic Dynamic Programming, Value Iteration, Policy Iteration
   - Structural Results: Supermodularity and monotone policies
6. **Discounted Cost POMDPs** [4 hours]
   - POMDP models and examples
   - Belief state formulation
   - Stochastic Dynamic Programming

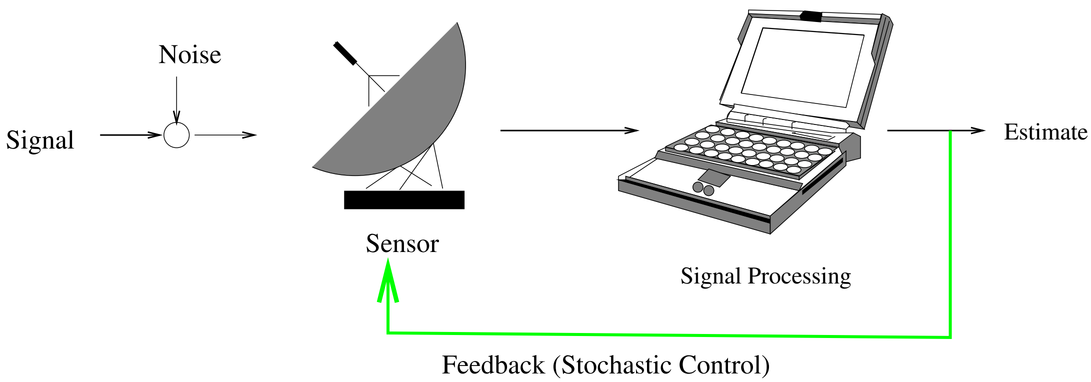
7. **Structural Results for POMDPs [9 hours]**

- Stochastic Orders
- Stochastic Dominance of Filters
- Lattice Programming
- Example 1: Quickest Detection with Optimal Sampling
- Example 2: Optimized Social Learning
- Example 3: Global Games
- Multi-armed bandits.
- Stochastic Gradient and Reinforcement Learning Algorithms

Applications



Sensor Adaptive Signal Processing – Active Sensing



Key idea: Feedback and Reconfigurability leading to a smart sensor – "controlled sensing"
Key Issue: Dynamic Decision making under uncertainty - partially observed Markov decision process.

Applications

1. Social/Telecommunication Networks: How to estimate sentiment in a social network? How to estimate degree distribution from a sampled evolving graph? Control of networks.

2. Sensor and Radar Signal Processing: How to control a sensing system? Radar tracking of ships, aircraft. Sonar tracking of submarines, surveillance.

3. Seismology and Geophysics

4. Mathematical finance and econometrics

5. Biomedical signal processing: genomic signal processing, biosensors, etc

6. Interacting Decision Makers and Social Learning in micro-economics

---

Quickest Detection and other Sequential Detection problems are special cases of POMDPs.

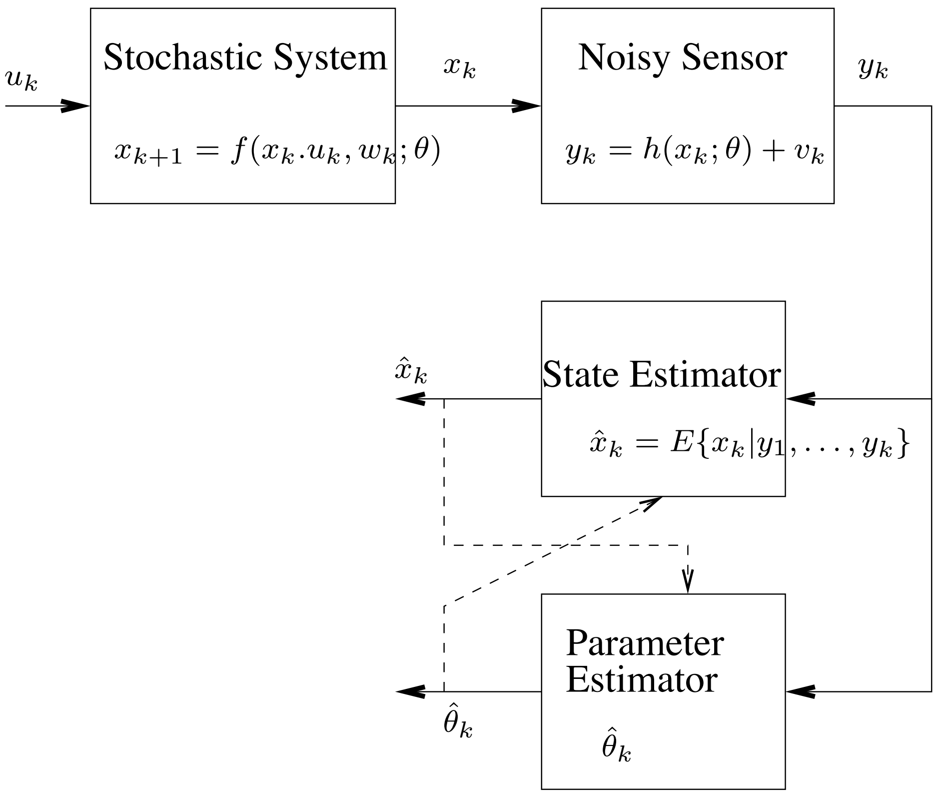
POMDPs are used extensively in robot navigation and planning. Also in Dialog Systems

Basic Setup

Partially observed stochastic dynamical system.

Known input u_k Observation y_k

Parameter θ state x_k

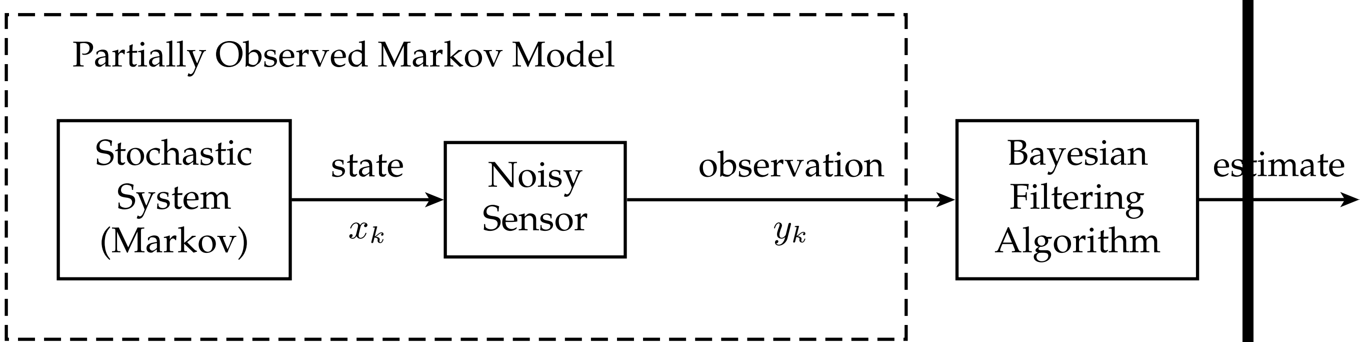


Q1: State Estimation **Q2**: Parameter Estimation

Q3: Controlled Sensing

Part 1. Bayesian Filtering

Applications: machine learning, robotics, systems & control, navigation, defense...



Ex 1: Linear State Estimation Problem Target evolves in 2 dimensional space – e.g. ship

$$x_{k+1} = Ax_k + Bu_k + w_k$$

$x[k] \stackrel{\triangle}{=} [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]'$ is target state vector
Noisy observations are obtained at sensor

$$y_k = Cx_k + v_k$$

where v_k denotes measurement noise.

Aim: Design a real time filtering algorithm – to estimate x_k given measurements $y_1, \ldots, y_k$.

Ex 2: Nonlinear Filtering. Localization Problem based on measured angles

Typical target model: $x_k = [r_k^x, r_k^y, v_k^x, v_k^y]'$

$$x_k = A_k x_{k-1} + w_k, \quad w_k \sim N(0, \sigma_w^2)$$

Measured data: noisy measurement of angle

$$\begin{aligned} y_k &= \arctan\left(\frac{r_k^x}{r_k^y}\right) + v_k \\ &= f(x_k) + v_k, \quad v_k \sim N(0, \sigma_v^2) \end{aligned}$$

Aim: Estimate x_k given observation history $y_1, \ldots, y_k$ in a recursive manner.

Target estimation is a filtering problem. The optimal state estimator – *filter* for this problem is not finite dimensional.

Ex 3: Maneuvering Targets

$$x_{k+1} = Ax_k + Bs_k + w_k$$
$$y_k = Cx_k + v_k$$

$s_k = (s_k^{(x)}, s_k^{(y)})'$ is maneuver command. s_k is modelled as a finite state Markov chain. This model is a *Jump Markov Linear System* (JMLS).

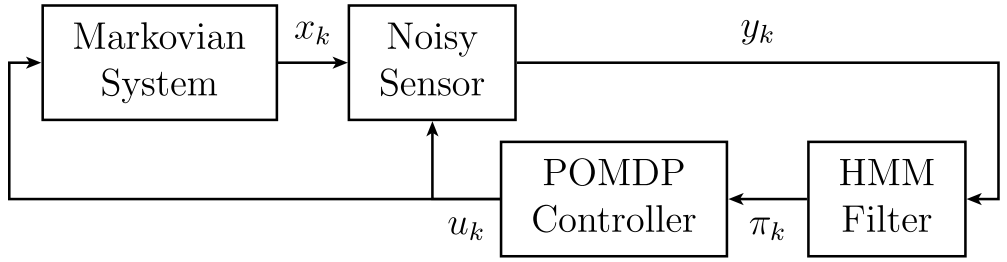
Note: HMMs and Linear State Space models are special cases of JMLS. State estimation of JMLS is surprisingly difficult – computing the optimal state estimate is exponentially hard. Numerous suboptimal state estimation algorithms e.g., IMM, Particle filters.

A more general version of a JMLS is

$$x_k = A(s_k)x_{k-1} + B(s_k)u_k + G(s_k)v_k$$
$$y_k = C(s_k)x_k + D(s_k)w_k + H(s_k)u_k$$

Here s_k is a finite state Markov chain, u_k denotes a known (exogeneous) input. x_k is the continuous valued state.

Part II: Partially Observed Stochastic Control



Ex 4: Optimal Observer Trajectory

Suppose observer can move. The model is

$$x_k = A_k x_{k-1} + w_k$$

$$y_k = f(x_k, u_k) + v_k$$

What is its optimal trajectory $\{u_k\}$*?*

Compute optimal trajectory $\{u_k\}$ to minimize

$$J = \sum_{k=1}^{N} \mathbf{E}\left\{[x_k - \hat{x}_k(u_k)]^2\right\}$$

This is a *partially observed stochastic control* problem called the *sensor scheduling problem.* Another possible cost: mutual information.

Intelligent Target Tracking

Suppose target is aware of observer.
Target maneuvers based on trajectory of observer.
The model is

$$x_k = A_k x_{k-1} + w_k + u_k^P$$
$$y_k = f(x_k, u_k^M) + v_k$$

What is optimal trajectory of observer $\{u_k^M\}$ and optimal control for target $\{u_k^P\}$?

This is a "full blown" control-scheduling problem. In the third part we address such problems.

Such partially observed stochastic control problems require state estimation as an integral part of the solution.

Controlled Sensing problems are special cases of POMDPs

Structure of Problem

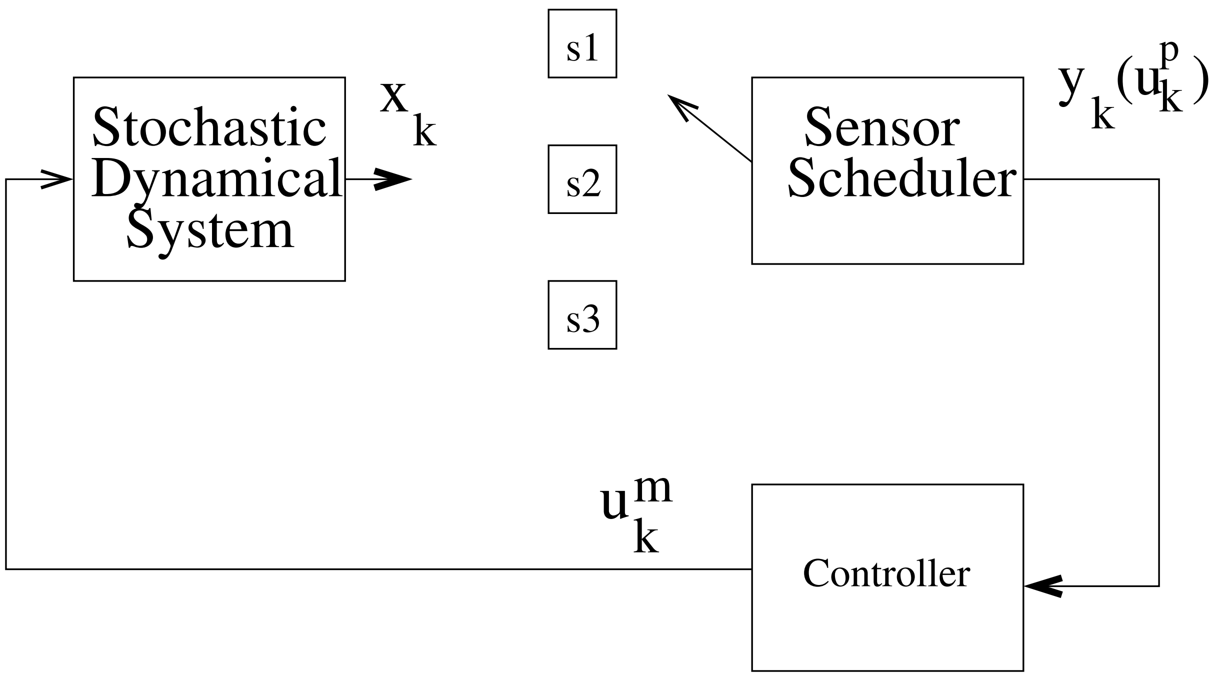
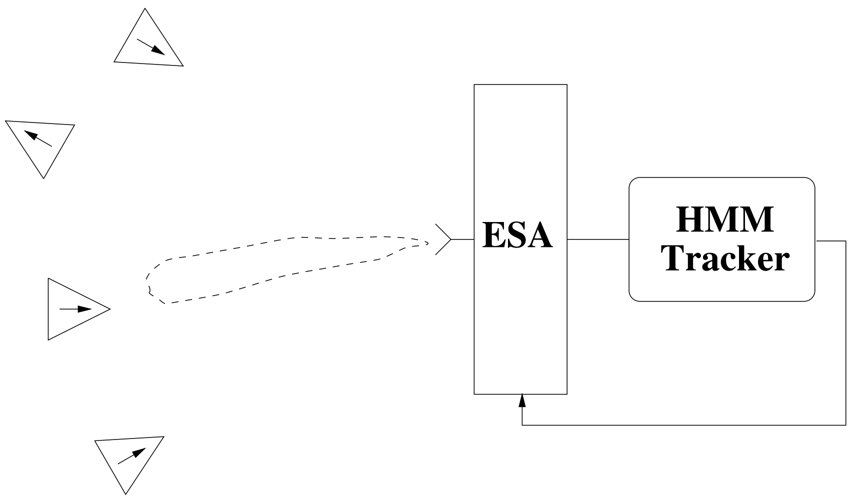


Figure 1: Sensor Scheduling and State Control

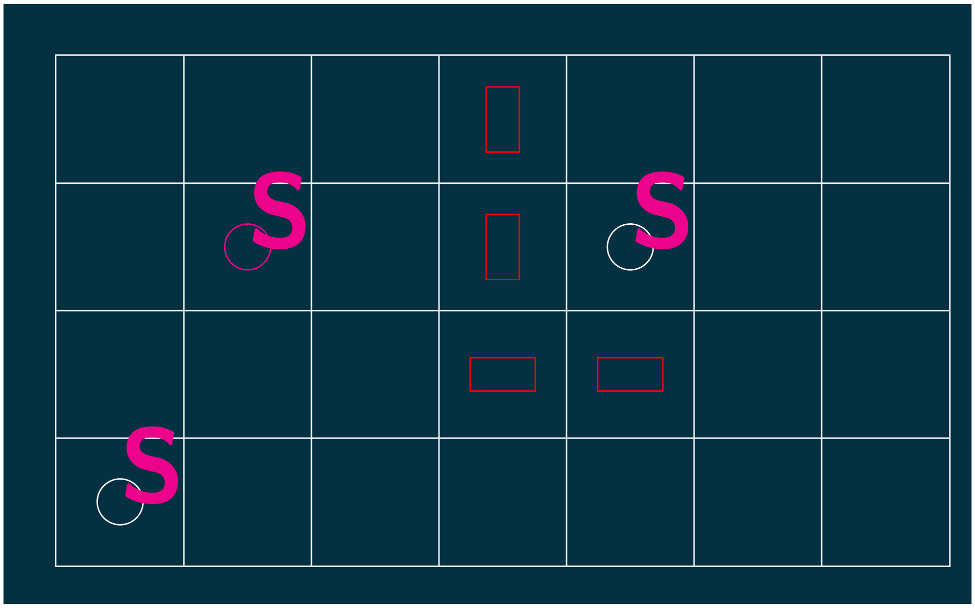
Controlled Sensing Problem

Example 1: Smart (Cognitive) Radar



Which target should the radar look at?

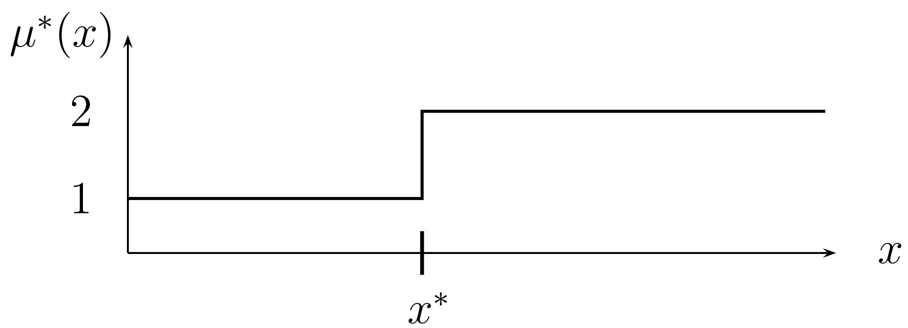
Example 2: Optimal Search/Multiarm Bandits



Structural Results

POMDPs suffer from the curse of dimensionality – exponential computational cost and memory (PSPACE hard).

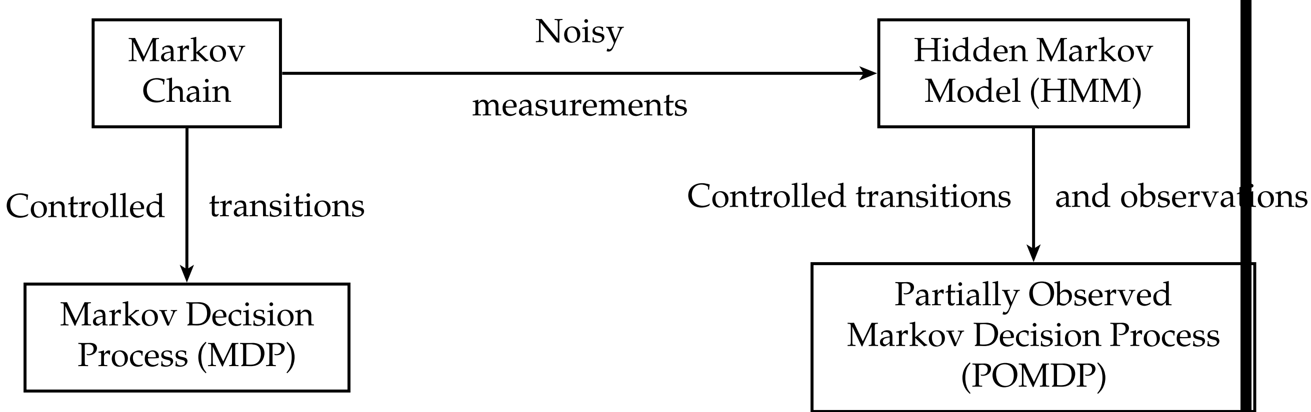
Structural results: Are there sufficient conditions on POMDP model so that optimal policy has "simple" structure?
Supermodularity, lattice programming, Monotone Comparative Statics: see Topkis book [1998].
Under what conditions of f does $u^*(x) = \text{argmax}_u f(x,u) \uparrow x$*?*



Monotone threshold policy

Then use machine learning algorithm to estimate optimal policy

Big Picture



What courses can you do after this course?

Dynamical Games, Deeper ideas in stochastic convergence of algorithms, stochastic calculus (continuous-time), more advanced concepts in reinforcement learning