1 Course overview

The course outline is available at `http://black.csl.uiuc.edu/~meyn/pages/ECE555-2008`
In short, there are four parts,

I Markov Models	II Optimal Control
III Linear Theory	IV Adaptation and Learning.

In this first lecture we survey some of the key concepts to be covered in the course.

1.1 Markov models

We let $\boldsymbol{X} = \{X(0), X(1), X(2), \dots\}$ denote the "state process" for a Markov model. For each time $t \ge 0$, it is assumed that $X(t)$ takes values in a state space denoted X. The state space will be a subset of Euclidean space $\mathbb{R}^n$ for some $n \ge 1$.

Recall that $\boldsymbol{X}$ is Markov provided the "memoryless property" holds. We describe this as follows: For any open set $O \subset \mathsf{X}$, any time t, and any initial $X(0)$,

$$\mathsf{P}\{X(t+1) \in O \mid X(0), \dots, X(t)\} = \mathsf{P}\{X(t+1) \in O \mid X(t)\} \tag{1}$$

We always assume that $\boldsymbol{X}$ has *stationary increments*, which means that the right hand side of (1) is independent of t. We let P denote the *transition kernel*,

$$P(x, O) = \mathsf{P}\{X(t+1) \in O \mid X(t) = x\}, \qquad x \in \mathsf{X},\ O \subset \mathsf{X}. \tag{2}$$

If $h \colon \mathsf{X} \to \mathbb{R}$ is a continuous function, it follows that the conditional expectation is also expressed in terms of P:

$$\begin{aligned} \mathsf{E}[h(X(t+1)) \mid X(0), \dots, X(t-1); X(t) = x] &= \mathsf{E}[h(X(t+1)) \mid X(t) = x] \\ &= \int P(x, dx_1) h(x_1) \end{aligned} \tag{3}$$

We restrict to open sets and continuous functions in these definitions only to avoid discussion of "measurability". Throughout most of the course we restrict to countable state spaces, in which case no restrictions are placed on the set O or function h. If X is finite, of size m, then P is interpreted as an $m \times m$ matrix. In this case $P(x_0, x_1)$ is the probability of moving from x_0 to x_1 in one time-step. The conditional expectation is expressed as a sum,

$$\mathsf{E}[h(X(t+1)) \mid X(t) = x] = \sum_{x_1 \in \mathsf{X}} P(x, x_1) h(x_1)$$

1.1.1 Examples

We begin with three basic examples. The linear model is the standard model in physics, systems theory, economics, and many other areas.

Example 1.1. ***The Linear State Space Model***

Suppose $\boldsymbol{X} = \{X(k)\}$ is a stochastic process for which there is an $n \times n$ matrix F and an i.i.d. sequence $\boldsymbol{\mathcal{E}}$ taking values in $\mathbb{R}^n$ such that the sequence of state values satisfies the recursion,

$$X(k+1) = FX(k) + \mathcal{E}(k+1), \qquad k \in \mathbb{Z}_+$$

where $X(0) \in \mathbb{R}$ is independent of $\boldsymbol{\mathcal{E}}$. Then $\boldsymbol{X}$ is called the (uncontrolled) linear state space model.

Its transition kernel is easily described: For any $x \in \mathsf{X} = \mathbb{R}^n$ and any set $O \subset \mathsf{X}$ we have,

$$P(x, O) = \mathsf{P}\{X(1) \in O \mid X(0) = x\} = \mathsf{P}\{Fx + \mathcal{E}(1) \in O\}$$

If in particular $\mathcal{E}(1)$ is Gaussian $N(0, \Sigma)$, then $P(x, \cdot\,)$ is also a Gaussian distribution, but with mean Fx rather than zero.

We also consider the linear state space model without noise: We denote the process using a lower case variable,

$$x(k+1) = Fx(k), \qquad k \geq 0. \tag{4}$$

The deterministic model is also Markovian: even if we know all of the values of $\{x(t), t \leq k\}$ then we will still predict $x(k+1)$ in the same way, with the same (exact) accuracy, based solely on (4) which uses only knowledge of $x(k)$.

Trajectories of the model with and without noise are shown in Figure 1. The common choice of F is $F = I + \Delta A$ with I equal to a 2×2 identity matrix, $A = \binom{-0.2,\ \ 1}{-1,\ -0.2}$ and $\Delta = 0.02$. The figure on the left shows a trajectory of the model without noise. The trajectory spirals towards the origin, and is intuitively "stable". For the model with noise, the common distribution of $\mathcal{E}(t)$ was taken Gaussian, of the form $\mathcal{E}(t) = HW(t)$ with $W(t)$ scalar $N(0,1)$ and $H = \binom{2.5}{2.5}$.

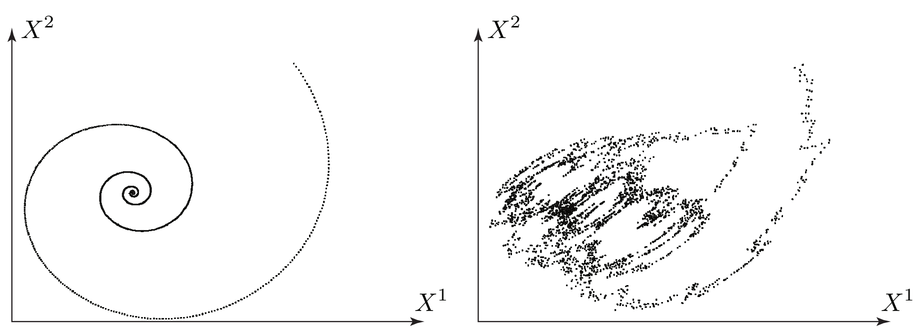

Figure 1: At left is a sample path generated by the deterministic linear model on $\mathbb{R}^2$. At right is a sample path from the linear state space model on $\mathbb{R}^2$ with Gaussian noise.

We will see that, in wide generality, if the deterministic linear model is stable, and the distribution of $\mathcal{E}(t)$ is "reasonable", then the linear state space model with non-zero noise is also stable in a stochastic sense. Generalizations of this principle hold for more general nonlinear models. ∎

This procedure can be generalized: A Markov model can be constructed as a nonlinear system perturbed by i.i.d. noise.

Example 1.2. ***The Nonlinear Linear State Space Model***

Suppose that $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuous, and that $\boldsymbol{\mathcal{E}}$ is i.i.d., taking values in $\mathbb{R}^m$. The nonlinear linear state space model is defined by the recursion,

$$X(k+1) = f(X(k), \mathcal{E}(k+1)), \qquad k \in \mathbb{Z}_+$$

where $X(0) \in \mathbb{R}$ is independent of $\boldsymbol{\mathcal{E}}$. For any $x \in \mathsf{X} = \mathbb{R}^n$ and any set $O \subset \mathsf{X}$ we have,

$$P(x, O) = \mathsf{P}\{X(1) \in O \mid X(0) = x\} = \mathsf{P}\{f(x, \mathcal{E}(1)) \in O\}$$

■

Random walks are defined by taking successive sums of independent and identically distributed (i.i.d.) random variables.

Example 1.3. ***Random Walks***

Suppose that $\boldsymbol{X} = \{X(k); k \in \mathbb{Z}_+\}$ is a sequence of random variables defined by,

$$X(k+1) = X(k) + \mathcal{E}(k+1), \qquad k \in \mathbb{Z}_+$$

where $X(0) \in \mathbb{R}$ is independent of $\boldsymbol{\mathcal{E}}$, and the sequence $\boldsymbol{\mathcal{E}}$ is i.i.d., taking values in $\mathbb{R}$. Then $\boldsymbol{X}$ is called a *random walk* on $\mathbb{R}$. The random walk is a special case of the one-dimensional linear state space model in which $F = 1$.

Suppose that the stochastic process $\boldsymbol{X}$ is defined by the recursion,

$$X(k+1) = [X(k) + \mathcal{E}(k+1)]_+ := \max(0, X(k) + \mathcal{E}(k+1)), \qquad k \in \mathbb{Z}_+,$$

where again $X(0) \in \mathbb{R}$, and $\boldsymbol{\mathcal{E}}$ is an i.i.d. sequence of random variables taking values in $\mathbb{R}$. Then $\boldsymbol{X}$ is called the *reflected random walk*. The reflected random walk is a special case of the one-dimensional nonlinear linear state space model in which $f(x, e) = [x + e]_+$ for each $x, e \in \mathbb{R}$.

■

The reflected random walk is both a model for storage systems and a model for queueing systems. For all such applications there are similar concerns and concepts of the structure and the stability of the models: we need to know whether a dam overflows, whether a queue ever empties, whether a computer network jams.

In fact, a standard queueing model is one of the simplest examples of a Markov chain, and is also an example of the reflected random walk.

Example 1.4. ***The M/M/1 queue***

The transition function for the *M/M/1 queue* is defined as

$$\mathsf{P}(Q(t+1) = y \mid Q(t) = x) = P(x, y) = \begin{cases} \alpha & \text{if } y = x + 1 \\ \mu & \text{if } y = (x-1)_+, \end{cases} \tag{5}$$

where α denotes the arrival rate to the queue, μ is the service rate, and these parameters are normalized so that $\alpha + \mu = 1$.

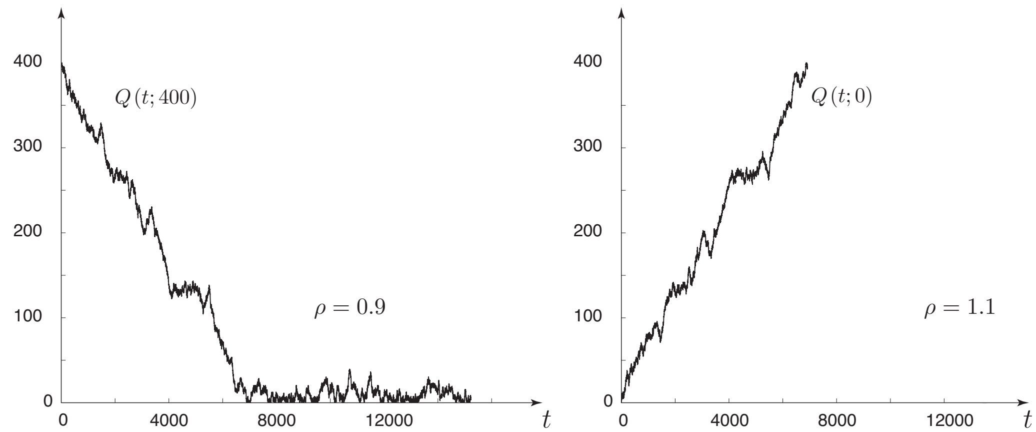

Figure 2: The M/M/1 queue: In the stable case on the left we see that the process $Q(t)$ appears piecewise linear, with a relatively small high frequency 'disturbance'. The process explodes linearly in the unstable case shown at right.

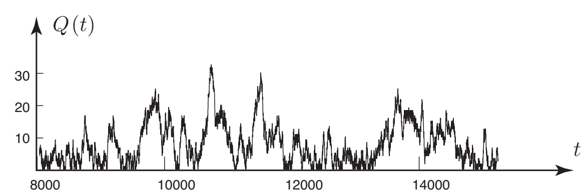

Figure 3: A close-up of the trajectory shown on the left hand side of Figure 2 with load $\rho = 0.9 < 1$. After a transient period, the queue length oscillates around its steady-state mean of 9.

The parameter $\rho := \alpha/\mu$ is known as the *load* for the queue. If $\rho := \alpha/\mu < 1$ then the arrival rate is strictly less than the service rate. In this case the process is ergodic: there is a probability measure π such that for any initial queue length $Q(0)$, and any integer $m \geq 0$,

$$\lim_{t\to\infty} \mathsf{P}\{Q(t) = m\} = \pi(m)$$

The probability can be identified as geometric with parameter ρ, so that $\pi(m) = (1-\rho)\rho^m$. The existence of an invariant measure π is interpreted as a form of stability for the queueing model, so that the sample path behavior looks like that shown in the left hand side of Figure 2 and in Figure 3. ∎

1.1.2 Value functions

Example 1.5. ***Value functions for the LSS Model***

Let $Q \geq 0$ be a positive semi-definite $n \times n$ matrix, and interpret the quadratic $c(x) = \frac{1}{2}x^{\intercal}Qx$ as a *cost function*. For the deterministic model (4) the *total cost* is expressed,

$$J(x) = \sum_{t=0}^{\infty} c(x(t)), \qquad x(0) = x \in \mathbb{R}^n.$$

The function J is an example of a value function for the deterministic model. It satisfies a dynamic programing equation,

$$J(x) = c(x) + J(x(1)), \qquad x(0) = x,\ x(1) = Fx$$

This is seen by writing $J(x) = c(x) + \sum_{t=1}^{\infty} c(x(t))$. In fact, provided J is finite-valued, it can be expressed as a quadratic $J(x) = \frac{1}{2}x^{\mathrm{T}}Mx$. The dynamic programing equation becomes,

$$\tfrac{1}{2}x^{\mathrm{T}}Mx = \tfrac{1}{2}x^{\mathrm{T}}Qx + \tfrac{1}{2}(Fx)^{\mathrm{T}}MFx, \qquad x \in \mathbb{R}^n$$

Hence, M solves the Lyapunov equation,

$$M = Q + F^{\mathrm{T}}MF \tag{6}$$

The function J also solves a dynamic programming equation for the model with noise. Suppose that $\boldsymbol{\mathcal{E}}$ is i.i.d. with zero mean, and finite second moment. We then have,

$$\mathsf{E}[J(X(1)) \mid X(0) = x] = \mathsf{E}[\tfrac{1}{2}(Fx + \mathcal{E}(1))^{\mathrm{T}}M(Fx + \mathcal{E}(1))]$$

Expanding the sum, and using the assumption that $\mathsf{E}[\mathcal{E}(1)] = 0$, we obtain,

$$\mathsf{E}[J(X(1)) \mid X(0) = x] = \tfrac{1}{2}(Fx)^{\mathrm{T}}MFx + \eta$$

with $\eta = \mathsf{E}[\frac{1}{2}(\mathcal{E}(1))^{\mathrm{T}}M\mathcal{E}(1)]$. From the Lyapunov equation (6) we obtain $\frac{1}{2}(Fx)^{\mathrm{T}}MAx = \frac{1}{2}x^{\mathrm{T}}(M - Q)x$. We conclude,

$$\mathsf{E}[J(X(1)) \mid X(0) = x] = J(x) - c(x) + \eta \tag{7}$$

The identity (7) is known as *Poisson's equation* for the Markov model, with *forcing function* c. We will see that η is the steady-state cost,

$$\eta = \lim_{t\to\infty} \mathsf{E}[c(X(t)) \mid X(0) = x], \qquad x \in \mathbb{R}^n. \tag{8}$$

The conclusion that the total cost J serves as a value function for the stochastic and deterministic models is a theme in this course. In nonlinear settings a deterministic model often provides structural insight that can be used to approximate a value function for the Markov model. ■

1.2 Optimization

Suppose now that a control sequence $\boldsymbol{U}$ is introduced, and that the cost function may depend on both the control and the state. We illustrate the issues to be addressed using the linear state space model.

Example 1.6. ***Optimization of the LSS Model***

The controlled linear model is defined as in Example 1.1. Suppose that $\boldsymbol{U}$ evolves on $\mathbb{R}^m$, and let G denote an $n \times m$ matrix. The controlled linear state space model is defined by the recursion,

$$X(k+1) = FX(k) + GU(k) + \mathcal{E}(k+1), \qquad k \in \mathbb{Z}_+ \tag{9}$$

It is again assumed that $X(0) \in \mathbb{R}$ is independent of $\boldsymbol{\mathcal{E}}$. Moreover, for each t it is assumed that $\{U(k) : k \leq t\}$ is independent of $\{\mathcal{E}(k) : k > t\}$. We usually assume that $\boldsymbol{U}$ is adapted to $\boldsymbol{X}$: For each t, there is a function ϕ such that,

$$U(t) = \phi(X(0), \dots, X(t), t).$$

Let $Q \geq 0$ by a positive semi-definite $n \times n$ matrix, $R > 0$ a positive definite $m \times m$ matrix, and define the quadratic cost by $c(x, u) = \frac{1}{2}(x^{\mathrm{T}} Q x + u^{\mathrm{T}} R u)$. For the model without noise we defined the (optimal) value function by,

$$J^*(x) = \min_{\boldsymbol{u}} \Big[\sum_{t=0}^{\infty} c(x(t), u(t)) \Big], \qquad x(0) = x \in \mathbb{R}^n,$$

where the minimum is over all adapted $\boldsymbol{u}$. We again have a dynamic programming equation,

$$J^*(x) = \min_{u} \Big[c(x, u) + J^*(x(1)) \Big], \qquad x(0) = x \in \mathbb{R}^n, \tag{10}$$

where $x(1) = Fx + Bu$ depends on u.

We will see that J^* is again quadratic, $J(x) = \frac{1}{2} x^{\mathrm{T}} M x$ for $x \in \mathbb{R}^n$, and that the optimal control is state feedback $u(t) = Kx(t)$ for some $m \times n$ matrix K (determined from M through the minimization (10)).

And, the value function solves a dynamic programming equation for the model with noise: Exactly as in (7) we have,

$$\min_{u} \mathsf{E}[c(X(0), U(0)) + J^*(X(1)) \mid X(0) = x,\ U(0) = u] = J^*(x) + \eta^* \tag{11}$$

The constant η^* is the minimal average cost, minimizing the limit (8) over all adapted $\boldsymbol{U}$. ■

We will develop dynamic programming equations for general Markov models, and consider deterministic analogs to obtain structural insights. We will also consider control of *partially observed* models in which $\boldsymbol{X}$ is not directly observed.

1.3 Linear theory

The development of the Lyapunov equation and Riccatti equation is identical to the continuous time construction seen in, for example, ECE 515 — if these terms aren't familiar to you, don't worry! For the linear state space model with Gaussian noise the partially observed control problem is solved using the Kalman filter.

1.4 Adaptation & learning

Learning in this course means finding the best approximation of a value function or a system model over a given class. Moreover, the approximation is based on observations of the system of interest, perhaps while the system is being controlled. Four instances of learning are,

(i) *Simulation*: To estimate the average cost (8) we can construct sample path averages from computer simulation,

$$\widehat{\eta}(n) = \frac{1}{n}\sum_{t=0}^{n-1} c(X(t)), \qquad n \geq 1.$$

Simulation is a special case of the stochastic approximation algorithm of Robbins and Monro, which is a basis of the learning mechanisms to be developed in the next three examples.

(ii) *Stochastic adaptive control*: What if the matrices (F, G) appearing in (9) are not known? These parameters can be estimated using methods similar to simulation.

(iii) Q-*learning*: The controlled Markov model is based on a controlled transition kernel denoted P_u, where u varies over possible control values. The Q-learning algorithm learns the dynamics defined by P_u, and simultaneously learns an optimal policy.

(iv) *TD-learning*: A method to learn optimal approximations of value functions for a Markov model.