Math 525: Lecture 21

March 29, 2018

So far, we have worked primarily with (stationary) Markov chains whose transition matrices are "constant". In this lecture, we explore the following question: what if we could "control" the transition matrix? In this context, we will have a transition matrix $P(\pi)$ that depends on some quantity π which we, the "controller", get to choose.

1 Markov decision processes

For this lecture, our setting is as follows:

- $S = \{1, \ldots, m\}$ is a finite state space.
- To each state i in S is associated a nonempty countable set $\mathcal{A}_i$ which we can intuitively think of as all the "actions" available at state i.

Definition 1.1. A *stationary policy* π_0 is a function whose domain is S and which satisfies $\pi_0(i) \in \mathcal{A}_i$ for all i. The set of all stationary policies is denoted Π_0

Definition 1.2. A *randomized policy* $(\pi_n)_{n\geq 0}$ is a sequence in which each $\pi_n(i)$ is a random variable satisfying

- $\pi_n(i)$ takes values in $\mathcal{A}_i$ a.s. and
- $\{\pi_n(i) = a\} \in \mathcal{F}_n$ for each a in $\mathcal{A}_i$.

We have purposely neglected to specify $\mathcal{F}_n$ yet (we'll come back to this). The set of all randomized policies is denoted Π.

For each state i in S and action a in $\mathcal{A}_i$, let $p_i(a)$ denote a nonnegative column vector satisfying $p_i(a)^\intercal e = 1$. Given a randomized policy π, let $(X_n^\pi)_{n\geq 0}$ denote a Markov chain satisfying

$$\mathbb{P}(X_{n+1}^\pi = j \mid X_n^\pi = i) = p_i(\pi_n(i))^\intercal e_j.$$

That is, the transition matrix at time n is

$$P(\pi_n) = \begin{pmatrix} p_1(\pi_n(1))^\intercal \\ p_2(\pi_n(2))^\intercal \\ \vdots \\ p_m(\pi_n(m))^\intercal \end{pmatrix}.$$

While we write X_n^π, it should be noted that this quantity only depends on $\pi_0, \dots, \pi_{n-1}$. In light of this, we sometimes write X_0, $X_1^{\pi_0}$, $X_2^{\pi_0,\pi_1}$, etc. to stress this independence. In light of this, we can unambiguously define $\mathcal{F}_n = \sigma(X_0^\pi, \dots, X_n^\pi)$.

Now, let $c : S \to \mathbb{R}$, $0 \le d < 1$, and

$$J(i,\pi) = \mathbb{E}\left[\sum_{n\ge 0} d^n c(X_n^\pi) \middle| X_0^\pi = i\right]. \tag{1}$$

We can think of

- $c(X_n^\pi)$ as the cost incurred at time n and
- d^n as a discount factor which attempts to capture the fact that costs incurred in the "future" are not as bad as costs incurred "today".

Our objective is to pick π so as to minimize $J(i,\pi)$. That is, we are interested in the quantity

$$\boxed{v(i) = \inf_{\pi\in\Pi} J(i,\pi)} \tag{2}$$

We call (2) a *Markov decision process* (MDP).

Proposition 1.3. *$v(i)$ is bounded for each i.*

Proof. This is a trivial consequence of the discount factor being strictly less than one:

$$|v(i)| \le \sum_{n\ge 0} d^n \max_j |c(j)| = \frac{1}{1-d}\max_j |c(j)|. \qquad \square$$

2 Dynamic programming

By the Markov property,

$$\begin{aligned} J(i,\pi) = \mathbb{E}^i\left[c(X_0^\pi) + \sum_{n\ge 1} d^n c(X_n^\pi)\right] &= c(i) + d\mathbb{E}^i\left[\sum_{n\ge 0} d^n c(X_{n+1}^\pi)\right] \\ &= c(i) + d\mathbb{E}^i\left[J(X_1^\pi, (\pi_n)_{n\ge 1})\right] \ge c(i) + d\sum_j (P(\pi_0))_{ij} v(j) \end{aligned}$$

where π_0 is some stationary policy. Taking infimums of both sides of this equality,

$$v(i) \ge \inf_{\pi_0\in\Pi_0}\left\{c(i) + d\sum_j (P(\pi_0))_{ij} v(j)\right\}. \tag{3}$$

Now, fix $\epsilon > 0$. For each i, let $\pi^i = (\pi_n^i)_{n\ge 0}$ be a randomized policy which satisifes

$$v(i) \ge J(i,\pi^i) + \epsilon.$$

Let π_0 be an arbitrary stationary policy. Define a new randomized policy $\pi^\epsilon = (\pi_n^\epsilon)_{n\geq 0}$ by

$$\pi_n^\epsilon = \begin{cases} \pi_0 & \text{if } n = 0 \\ \mathbf{1}_{\{i\}}(X_1^{\pi_0})\pi_{n-1}^i & \text{if } n > 0. \end{cases}$$

Note that

$$v(i) \leq J(i,\pi^\epsilon) = c(i) + d\sum_j (P(\pi_0))_{ij} J(j,\pi^j) \leq c(i) + d\sum_j (P(\pi_0))_{ij} v(j) - \epsilon.$$

Now, take infimums of both sides to get

$$v(i) \leq \inf_{\pi_0 \in \Pi_0} \left\{ c(i) + d\sum_j (P(\pi_0))_{ij} v(j) \right\} - \epsilon. \tag{4}$$

We can take $\epsilon \downarrow 0$ and combine (3) and (4) to arrive at

$$v(i) = \inf_{\pi_0 \in \Pi_0} \left\{ c(i) + d\sum_j (P(\pi_0))_{ij} v(j) \right\}. \tag{5}$$

The implications of this are amazing! We started out with an objective function (1) that was daunting: minimizing it would require picking a stationary policy for each time n. However, we were able to use the Markov property to reduce this to a "local" problem that only involves minimizing over all stationary policies π_0. In fact, we can simplify (5) even further. First, we need some notation:

$$\text{for } \{y_\alpha\}_\alpha \in \mathbb{R}^n, \ \inf_\alpha y_\alpha \text{ is the vector with entries } \inf_\alpha (y_\alpha)_i.$$

Theorem 2.1 (Dynamic programming)**.** *Let $v = (v(1), \ldots, v(m))^\intercal$ and $c = (c(1), \ldots, c(m))^\intercal$ where $v(i)$ is the quantity defined by (2). Then,*

$$\boxed{\sup_{\pi_0 \in \Pi_0} \left\{ (I - dP(\pi_0))\, v - c \right\} = 0}$$

Proof. We can rewrite (5) as

$$v = \inf_{\pi_0 \in \Pi} \left\{ c + dP(\pi_0) v \right\}. \tag{6}$$

Moving some terms around, we obtain the desired result. □

In fact, the situation is much more general than we have let on. We can allow for more general discount factors and costs:

$$J(i,\pi) = \mathbb{E}\left[\sum_{n\geq 0} d(\pi_n, X_n^\pi)^n c(\pi_n, X_n^\pi) \middle| X_0^\pi = i \right].$$

However, in this case, it is no longer necessarily the case that $v(\cdot)$ is bounded. When it is, the corresponding dynamic programming equation is

$$\sup_{\pi_0 \in \Pi_0} \left\{ (I - D(\pi_0)P(\pi_0))\, v - c(\pi_0) \right\} = 0 \tag{7}$$

where $D(\pi_0) = \operatorname{diag}(d(\pi_0(1),1), \ldots, d(\pi_0(m),m))$ and $c(\pi_0) = (c(\pi_0(1),1), \ldots, c(\pi_0(m),m))^\intercal$.

In light of this, the remainder of this lecture is focused on (7). In particular, we would like to know if an arbitrary vector v satisfies (7), is it necessarily equal to the MDP (2)? Moreover, can we use (7) to compute the MDP?

3 Matrix classes

First, we will need to recall some more linear algebra.

3.1 Monotone matrices

Definition 3.1. A *monotone matrix* is a real square matrix A such that $Ax \geq 0$ implies $x \geq 0$ for all real vectors x.

Proposition 3.2. *Monotone matrices are nonsingular.*

Proof. Let A be a monotone matrix and assume there exists x with $Ax = 0$. Then, by monotonicity, $x \geq 0$ and $-x \geq 0$, and hence $x = 0$. □

Proposition 3.3. *A real square matrix A is monotone if and only if A^{-1} exists and is nonnegative.*

Proof. Suppose A is monotone. Denote by x the i-th column of A. Then, Ax is the i-th standard basis vector, and hence $x \geq 0$ by monotonicity. For the reverse direction, suppose A admits a nonnegative inverse. Then, if $Ax \geq 0$, $x = A^{-1}Ax \geq A^{-1}0 = 0$, and hence A is monotone. □

3.2 M-matrices

Definition 3.4. An *M-matrix* is any square matrix A which can be written in the form

$$A = sI - B \tag{8}$$

where $s \geq \rho(B)$ and B is nonnegative.

Proposition 3.5. *The M-matrix (8) is nonsingular if and only if $s > \rho(B)$.*

Proof. Note that A is nonsingular if and only if s is an eigenvalue of B since

$$Ax = sx - Bx = 0 \iff Bx = sx.$$

Therefore, if $s > \rho(B)$, then A is nonsingular. Conversely, if $s = \rho(B)$, by the Perron-Frobenius theorem, s is an eigenvalue of B. □

Proposition 3.6. *Nonsingular M-matrices are monotone.*

Proof. Divide (8) by s so that

$$s^{-1}A = I - s^{-1}B.$$

Noting that $\rho(s^{-1}B) < 1$, the inverse of the right hand side of the above is the Neumann series

$$\left(I - s^{-1}B\right)^{-1} = \sum_{k \geq 0} \left(s^{-1}B\right)^k.$$

In particular, this Neumann series consists only of powers of nonnegative matrices, and therefore converges to a nonnegative matrix. In other words, sA^{-1} is nonnegative, and hence so too is A^{-1}. □

Figure 1: Graph of (10)

3.3 Weakly chained diagonally dominant matrices

Definition 3.7. Let $A = (A_{ij})$ be a matrix. We say its i-th row is *strictly diagonally dominant* (s.d.d.) if

$$|A_{ii}| > \sum_{i \neq j} |A_{ij}| \,. \tag{9}$$

We say the matrix is s.d.d. if all of its rows are s.d.d. Weakly diagonally dominant (w.d.d.) is defined with weak inequality ($\geq$) instead.

Example 3.8. The matrix

$$\begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{pmatrix} \tag{10}$$

is not strictly diagonally dominant, but it is weakly diagonally dominant.

Definition 3.9. Let $A = (A_{ij})$ be a matrix. Let

$$J = \{i \colon i \text{ satisfies (9)}\}$$

denote the set of all s.d.d. rows of A. We say A is *weakly chained diagonally dominant* (w.c.d.d.) if

1. A is w.d.d.
2. For each row $i \notin J$, there is a walk $i \to j$ with $j \in J$.

Example 3.10. The matrix (10) is w.c.d.d. (see Figure 1).

Proposition 3.11. *w.c.d.d. matrices are nonsingular.*[1]

Proof. Let A be w.c.d.d. If A is singular, we can find a nonzero vector x such that $Ax = 0$. Without loss of generality, let i_1 be such that $|x_{i_1}| = 1 \geq |x_j|$ for all j. Since A is w.c.d.d., we may pick a walk $i_1 \dashrightarrow i_2 \dashrightarrow \cdots \dashrightarrow i_k$ ending at an s.d.d. row $i_k \in J$.

Taking moduli on both sides of

$$-a_{i_1 i_1} x_{i_1} = \sum_{j \neq i_1} a_{i_1 j} x_j$$

[1]Shivakumar, P. N., and Kim Ho Chew. "A sufficient condition for nonvanishing of determinants." Proceedings of the American mathematical society (1974): 63-66.

yields

$$|a_{i_1 i_1}| = |a_{i_1 i_1} x_{i_1}| = \left| \sum_{j \neq i_1} a_{i_1 j} x_j \right| \leq \sum_{j \neq i_1} |a_{i_1 j}| \, |x_j| \leq \sum_{j \neq i_1} |a_{i_1 j}| \, .$$

Since A is w.d.d., the above must hold with equality. Therefore, $|x_j| = 1$ whenever $a_{i_1 j}$ is nonzero. In particular, $|x_{i_2}| = 1$, and we can repeat the same argument as above to get that row i_2 is not s.d.d., row i_3 is not s.d.d., etc. until we conclude that row i_k is not s.d.d., a contradiction. □

To be continued...