Markov Decision Problem
Linear Programming Method

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

☞ Linear Programming Algorithm without Discounting

Optimizes the "average", i.e., expected, cost or return per period in steady state.

☞ Linear Programming Algorithm with Discounting

Optimizes the present value of all future expected costs

LP model of MDP

Assume that, using the optimal policy, a steady state distribution exists.

Define "randomized" or "mixed" strategies:

X_i^k = joint probability, in steady state, of being in state i and selecting action $k \in K_i$

LP Model

$$\text{Maximize} \sum_{i\in S} \sum_{k\in K_i} C_i^k X_i^k$$

$$\sum_{k\in K_j} X_j^k = \sum_{i\in S} \sum_{k\in K_i} p_{ij}^k X_i^k \quad \forall\, j\in S$$

$$\sum_{i\in S} \sum_{k\in K_i} X_i^k = 1$$

$$X_i^k \geq 0$$

One constraint is redundant, and can be eliminated.

Taxi Problem

Transition Probabilities

Action: Cruise

from \ to	1	2	3
1	0.5	0.25	0.25
2	0.5	0	0.5
3	0.25	0.25	0.5

Action: Cabstand

from \ to	1	2	3
1	0.0625	0.75	0.1875
2	0.0625	0.875	0.0625
3	0.125	0.75	0.125

Action: Wait for call

from \ to	1	2	3
1	0.25	0.125	0.625
2	0	1	0
3	0.75	0.0625	0.1875

Taxi Problem

Cost Matrix

k	name	1	2	3
1	Cruise	¯8	¯16	¯7
2	Cabstand	¯2.75	¯15	¯4
3	Wait for call	¯4.25	999	¯4.5

(Rows ~ actions, Columns ~ states)

A value of 999 above signals an infeasible action in a state.

Expected returns for each i&k

LP Tableau

Taxi Problem

									RHS
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	
Min	¯8	¯2.75	¯4.25	¯16	¯15	¯7	¯4	¯4.5	
	0.5	0.9375	0.75	¯0.5	¯0.0625	¯0.25	¯0.125	¯0.75	0
	¯0.25	¯0.75	¯0.125	1	0.125	¯0.25	¯0.75	¯0.0625	0
	1	1	1	1	1	1	1	1	1

Iteration 0

LP Tableau

Initial basic feasible solution

									RHS
basic:	★			★		★			
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	
Min	0	2.1	5.01667	0	¯4.95	0	¯0.566667	3.23333	9.2
	1	1.45	1.36667	0	0.35	0	0.0333333	¯0.616667	0.4
	0	¯0.4	0.1	1	0.3	0	¯0.4	0.15	0.2
	0	¯0.05	¯0.466667	0	0.35	1	1.36667	1.46667	0.4

Initial policy: in each city, select "cruise" ("greedy" policy)

Iteration 0

Policy: (Cost= ¯9.2)

State	Action	P{i}
1 Town A	1 Cruise	0.4
2 Town B	1 Cruise	0.2
3 Town C	1 Cruise	0.4

initial basic solution $\begin{cases} X_1^1 = 0.4 \\ X_2^1 = 0.2 \\ X_3^1 = 0.4 \end{cases}$

Initial policy: in each city, select "cruise" ("greedy" policy)

Iteration 0

LP Tableau

basic:	★			★		★			
k:	1	2	3	1	2	1	2	3	RHS
i:	1	1	1	2	2	3	3	3	
Min	0	2.1	5.01667	0	¯4.95	0	¯0.566667	3.23333	9.2
	1	1.45	1.36667	0	0.35	0	0.0333333	¯0.616667	0.4
	0	¯0.4	0.1	1	0.3	0	¯0.4	0.15	0.2
	0	¯0.05	¯0.466667	0	0.35	1	1.36667	1.46667	0.4

$$\text{minimum}\left\{\frac{0.4}{0.35}, \frac{0.2}{0.3}, \frac{0.4}{0.35}\right\} = \frac{0.2}{0.3}$$

X_2^2 enters the basis, replacing X_2^1

Iteration 1

LP Tableau

	★				★	★			
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	rhs
Min	0	¯4.5	6.6666	16.5	0	0	¯7.1666	5.70833	12.5
	1	1.9166	1.25	¯1.16667	0	0	0.5	¯0.79166	0.1666
	0	¯1.3333	0.3333	3.33333	1	0	¯1.3333	0.5	0.6666
	0	0.4166	¯0.5833	¯1.16667	0	1	1.8333	1.29167	0.1666

basic solution $\begin{cases} X_1^1 = 1/6 \\ X_2^2 = 2/3 \\ X_3^1 = 1/6 \end{cases}$

Iteration 1

Policy: (Cost= ¯12.5)

State	Action	P{i}
1 Town A	1 Cruise	0.166667
2 Town B	2 Cabstand	0.666667
3 Town C	1 Cruise	0.166667

Iteration 1

LP Tableau

	★				★	★			
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	rhs
Min	0	¯4.5	6.6666	16.5	0	0	¯7.1666	5.70833	12.5
	1	1.9166	1.25	¯1.16667	0	0	0.5	¯0.79166	0.1666
	0	¯1.3333	0.3333	3.33333	1	0	¯1.3333	0.5	0.6666
	0	0.4166	¯0.5833	¯1.16667	0	1	1.8333	1.29167	0.1666

$$\text{minimum}\left\{\frac{0.166}{0.5}, \frac{0.1666}{1.833}\right\} = \frac{0.1666}{1.8333}$$

X_3^2 enters the basis, replacing X_3^1

Iteration 2

LP Tableau

	★				★		★		
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	rhs
Min	0	¯2.8712	4.3863	11.9394	0	3.9090	0	10.7576	13.1515
	1	1.8030	1.4090	¯0.8484	0	¯0.2727	0	¯1.1439	0.1212
	0	¯1.0303	¯0.0909	2.4848	1	0.7272	0	1.4393	0.7878
	0	0.2272	¯0.3181	¯0.6363	0	0.5454	1	0.7045	0.0909

Note that for every state, there is a variable in the basis for only one action!

Iteration 2

Policy: (Cost= ¯13.1515)

State	Action	P{i}
1 Town A	1 Cruise	0.121212
2 Town B	2 Cabstand	0.787879
3 Town C	2 Cabstand	0.0909091

Iteration 2

LP Tableau

LP Tableau

		★			★		★		
k:	1	2	3	1	2	1	2	3	
i:	1	1	1	2	2	3	3	3	rhs
Min	1.59244	0	6.63025	10.5882	0	3.4747	0	8.9359	13.3445
	0.55462	1	0.78151	¯0.4705	0	¯0.1512	0	¯0.6344	0.06722
	0.57142	0	0.71428	2	1	0.5714	0	0.7857	0.85714
	¯0.12605	0	¯0.49579	¯0.5294	0	0.5798	1	0.8487	0.07563

Reduced costs are all nonnegative...
the optimality condition is satisfied!

Optimal Policy

Iteration 3

Policy: (Cost= ¯13.3445)

State	Action	P{i}
1 Town A	2 Cabstand	0.0672269
2 Town B	2 Cabstand	0.857143
3 Town C	2 Cabstand	0.0756303

The optimal policy found by the simplex LP algorithm is deterministic, not randomized, i.e., for each state, only one action is specified.

LP Algorithm for MDP with discounting

Determining a policy which minimizes the *present value* of all future costs over an infinitely long planning horizon.

Note: existence of a steady state distribution is *not* assumed!

The present value of future costs (i.e., the discounted future costs) will depend upon the initial state of the system.

Define

α_j = probability that system in initially in state j

Note: If the initial state is known, then $\alpha = [0, 0, \ldots, 0, 1, 0, \ldots 0]$

Decision variables

$\lambda_i^k(n)$ = Joint probability that system is in state j in period n and action $k \in K_j$ is selected

Note that this definition of the decision variables does not assume that the same policy is optimal for every stage!

Define

β = discount factor = $\frac{1}{1+r}$

where **r** = rate of return per stage

Then the present value of a cost Y which is incurred 1 period hence is βY

2 periods hence is $\beta^2 Y$

⋮

n periods hence is $\beta^n Y$

If C_j^k = cost of action k in state j

then

$$\sum_j \sum_{k \in K_j} C_j^k \lambda_j^k(n)$$ = expected cost during stage (period) n

and

$$\sum_{n=0}^{\infty} \beta^n \sum_j \sum_{k \in K_j} C_j^k \lambda_j^k(n)$$ = present value of all costs in periods n=0, 1, 2, ...

Our objective is therefore to minimize the discounted future expected costs:

$$\sum_j \sum_{k \in K_j} \left[\sum_{n=0}^{\infty} \beta^n C_j^k \lambda_j^k(n) \right]$$

Constraints

For each state j at stage n=0: $\sum_{k\in K_j} \lambda_j^k(0) = \alpha_j$

For each state j at stage n, n=1,2,...

$$\underbrace{\sum_{k\in K_j} \lambda_j^k(n)}_{\text{Probability that system is in state } j \text{ at stage } n} = \underbrace{\sum_i \sum_{k\in K_i} p_{ij}^k \lambda_i^k(n-1)}_{\text{Probability that system makes transition from state } i \text{ in stage } n-1 \text{ to state } j \text{ in stage } n}$$

Note that there is an infinite number of constraints, as well as infinitely many variables!

In order to reduce the size of the LP to finite proportions, we will utilize the ***z - transform.***

The z-transform of the sequence $\{a_n\}_{n=0}^{\infty}$ is the *function*

$$F(z) = \sum_{n=0}^{\infty} z^n a_n$$

[See Queueing Systems, Vol. I, Appendix I by L. Kleinrock]

Note that, given F, we can reconstruct the sequence:

$$a_n = \frac{1}{n!}\frac{d^nF(0)}{dz^n}$$

For each pair of state j and action k, consider the sequence of probabilities

$$\{\lambda_j^k(n)\}_{n=0}^{\infty}$$

Its z-transform is $F(z) = \sum_{n=0}^{\infty} z^n \lambda_j^k(n)$

Define a new set of decision variables

$$x_j^k = \sum_{n=0}^{\infty} \beta^n \lambda_j^k(n)$$

i.e., the z-transform of $\{\lambda_j^k(n)\}_{n=0}^{\infty}$ evaluated at β

We are then able to rewrite our objective function

$$\sum_j \sum_{k\in K_j} \left[\sum_{n=0}^{\infty} \beta^n C_j^k \lambda_j^k(n)\right]$$

with a finite number of terms:

$$\sum_j \sum_{k\in K_j} C_j^k x_j^k$$

where $x_j^k = \sum_{n=0}^{\infty} \beta^n \lambda_j^k(n)$

Constraints

In order to reduce the set of constraints to a finite number (with finitely many variables), perform the following operations:

- For each pair j & n, multiply the corresponding constraint by β^n

$$\begin{cases} \beta^0 \sum_{k\in K_j} \lambda_j^k(0) = \beta^0 \alpha_j & \text{for each state } j \\ \sum_{k\in K_j} \beta^n \lambda_j^k(n) = \beta \sum_i \sum_{k\in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1) & \text{for each state } j \text{ \& } n \ge 1 \end{cases}$$

- For each state j, sum the equations over n:

$$\begin{cases} \beta^0 \sum_{k\in K_j} \lambda_j^k(0) = \beta^0 \alpha_j & \text{for each state } j \\ \sum_{k\in K_j} \beta^n \lambda_j^k(n) = \beta \sum_i \sum_{k\in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1) & \text{for each state } j \text{ \& } n \ge 1 \end{cases}$$

$$\Longrightarrow \sum_{n=0}^{\infty} \sum_{k\in K_j} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_{n=1}^{\infty} \sum_i \sum_{k\in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

- Rearrange the order of summation in this new constraint:

$$\sum_{n=0}^{\infty} \sum_{k\in K_j} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_{n=1}^{\infty} \sum_i \sum_{k\in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

$$\Longrightarrow \sum_{k\in K_j} \sum_{n=0}^{\infty} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_i \sum_{k\in K_i} \sum_{n=1}^{\infty} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

$$\Longrightarrow \boxed{\sum_{k\in K_j} x_j^k = \alpha_j + \beta \sum_i \sum_{k\in K_i} x_i^k} \quad \text{for all } j$$

since $\sum_{n=1}^{\infty} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1) = \sum_{n=0}^{\infty} \beta^n \lambda_i^k(n)$

LP Model

Minimize $\sum_j \sum_{k\in K_j} C_j^k x_j^k$

subject to

$$\sum_{k\in K_j} x_j^k = \alpha_j + \beta \sum_i \sum_{k\in K_i} p_{ij}^k x_i^k \quad \text{for all } j$$

$$x_j^k \ge 0$$

Note that
- sum of x is not specified to be 1
- no redundant constraint was eliminated from state equations

Using the "Kronecker delta", i.e.,

$$\delta_{ij} \equiv \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i\neq j \end{cases}$$

this LP model may be rewritten:

$$\text{Minimize } \sum_j \sum_{k \in K_j} C_j^k x_j^k$$

subject to

$$\sum_i \sum_{k \in K_i} \left(\delta_{ij} - \beta\, p_{ij}^k\, x_i^k\right) = \alpha_j \quad \text{for all } j$$

$$x_j^k \geq 0$$

If x^* is the optimal basic solution, then

$x_j^{*k} > 0$ (i.e., basic)

implies that

the optimal policy is to select action k when in state j *for every stage n=0,1,2,....*

i.e., the optimal policy is stationary, same for every time period!