Final
Colorado CSCI 5454

Alex Book

December 9, 2021

Problem 1

Parts of DP solution

Let C be the table of dimension $n \times n$ holding solutions for our subproblems. Let $M_{i,j}$ be the set of coordinates of squares legally reachable from location (i, j). $r_{i,j}$ is the table holding the reward values for each square.

Subproblem definition

Subproblem (i, j) represents the maximum reward achievable when starting at location (i, j).

Recurrence

Base case(s):

$$C_{i,n} = r_{i,n} \text{ for all } i \in (1 \dots n)$$

Inductive case:

$$C_{i,j} = r_{i,j} + \max_{(k,\ell) \in M_{i,j}} \{C_{k,\ell}$$

Computing the final answer

To find the final answer, we simply take the maximum value subproblem from column 1. Since that is where the king is able to start, those are the subproblems that will hold our final answer.

Correctness:

The base case is simply representing that the squares the king ends on (those in the n^{th} column) must have a total value equal to their reward, as the king is both starting and ending on that square, not allowing any more value to be accrued.

For the inductive case, we start with the reward value of the current square. Since we're starting there, we accrue that value. Then, we look at what squares are reachable from the current square, and we add on whichever of those squares has the greatest total value (greatest reachable subproblem). We fill C starting from the right and moving leftward (solving all subproblems in one column before moving on to the column directly to the left), so the necessary subproblems will always be solved before they are needed.

Since the base cases are correct, so must be the subproblems in the column directly to the left of the n^{th} column (as those squares see the king move directly to the ending squares, which are the base case), and thus so are those in the column left of that one, so on and so forth until we reach column 1, where our final answer is found.

Problem 2

Part a

1.

$$\begin{bmatrix}1 & 2\\5 & -10\end{bmatrix} \cdot \begin{bmatrix}.5\\.5\end{bmatrix} = \begin{bmatrix}1.5\\-2.5\end{bmatrix}$$

2.

$$\begin{bmatrix}1 & 2\\5 & -10\end{bmatrix} \cdot \begin{bmatrix}.3\\.7\end{bmatrix} = \begin{bmatrix}1.7\\-5.5\end{bmatrix}$$

3.

$$\begin{bmatrix}1 & 2\\5 & -10\end{bmatrix} \cdot \begin{bmatrix}.7\\.3\end{bmatrix} = \begin{bmatrix}1.3\\.5\end{bmatrix}$$

Part b

Round 1

$w_1^1 = 1$
$w_2^1 = 1$
$p = \begin{bmatrix}.5\\.5\end{bmatrix}$
Weights and distribution are initialized to these values.

Round 2

$w_1^2 = 1(1 - .5(\frac{10-1}{20})) = .775$
$w_2^2 = 1(1 - .5(\frac{10-5}{20})) = .875$
$p = \begin{bmatrix}.47\\.53\end{bmatrix}$, distribution found through normalization

Round 3

$w_1^2 = .775(1 - .5(\frac{10-2)}{20})) = .62$
$w_2^2 = .875(1 - .5(\frac{10-(-10)}{20})) = .438$
$p = \begin{bmatrix}.586\\.414\end{bmatrix}$, distribution found through normalization

Part c

As $T \to \infty$, the algorithm's distribution over actions approaches $p = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. This makes sense, as the row player should optimally choose **go** every time since it yields a higher utility when the column player chooses **stop**. Each of the weights decreases over time, but the weight corresponding to **go** decreases more slowly, as its loss ($\frac{5}{20}$) is less than that of the loss corresponding to **stop** ($\frac{9}{20}$).

Problem 3

Let $x = x_1 \dots x_n$ represent how much of each item is purchased. $u = u_1 \dots u_n$ represents the utility of each item. $c = c_1 \dots c_n$ represents the cost of each item. $\ell = \ell_1 \dots \ell_n$ represents how much of each item must be bought (only applicable to part b). B represents your total budget.

Part a

Consider the following linear program:

$$\begin{aligned} \text{max. } & x^T u \\ \text{s.t. } & x^T c \leq B \\ & x \succeq 0 \end{aligned}$$

We are trying to maximize the total utility added up across all items, which is equivalent to the dot product of vectors x and u (one must be transposed for dimension compatibility). We are also faced with two constraints, the first of which is that we must not go over budget. Our total amount spent is equivalent to the dot product of vectors x and c. The second is that we cannot purchase negative amounts of products, which is equivalent to the vector inequality $x \succeq 0$.

Part b

Consider the following linear program:

$$\begin{aligned} \text{max. } & x^T u \\ \text{s.t. } & x^T c \leq B \\ & x \succeq \ell \end{aligned}$$

We are trying to maximize the total utility added up across all items, which is equivalent to the dot product of vectors x and u (one must be transposed for dimension compatibility). We are also faced with three constraints, the first of which is that we must not go over budget. Our total amount spent is equivalent to the dot product of vectors x and c. The second is that we cannot purchase negative amounts of products, and the third is that we must purchase at least ℓ_i amount of each item i. These last two constraints can be combined into one statement, since it is given that $\ell_i \geq 0$ for $i \in 1 \dots n$. They are equivalent to the vector inequality $x \succeq \ell$.