

# Approximate Dynamic Programming のノート

2013 年 5 月 24 日



## 第 1 章



## 第 2 章

# Some Illustrative Models

1. Deterministic Problems
2. Stochastic Problems  
the distribution is known.
3. Information acquisition problems  
the distribution is not known.

## 2.1 Deterministic Problems

### 2.1.1 The Shortest Path Problems

$I$  the set of nodes (intersections) in the network,

$L$  the set of links  $(i, j)$  in the network,

$c_{ij} > 0$  the cost (typically time) to drive from  $i$  to  $j$ ,  $i, j \in I, (i, j) \in L$ ,

$I_i^+$  the set of nodes  $j$  for which there is a link  $(i, j) \in L$ ,

$I_j^-$  the set of nodes  $i$  for which there is a link  $(i, j) \in L$ .

node  $i \in I$  にいる旅行者は,  $j \in I_i^+$  のノードに移動することが出来る. スタート node  $q$  から目的 node  $r$  に最小コストで移動するための経路は何か.

$v_j$  the minimum cost required to get from node  $j$  to  $r$ .

$v_j$  を求める.  $v_r = 0$  であることは分かる.  $v_j^n$  を  $n = 1, 2, \dots$  回目の繰り返しによる  $v_j$  の推定値としよう. 以下のアルゴリズムで  $v_j$  を求めることができる.

0.  $v_{j \neq r}^0 = M, v_r^0 = 0$ ,  $M$ : 十分大きな値,  $n = 1$  とする.
1. すべての  $i \in I$  について, 適当な順番で  $v_i^n$  を以下の式で求める.

$$v_i^n = \min_{j \in I_i^+} (c_{ij} + v_j^{n-1}) \quad (2.1)$$

$v_j^n$  が求まっていれば, 右辺の式中の  $v_j^{n-1}$  を置き換えて良い.

2.  $v_i^n < v_i^{n-1}$  なる  $i$  があるならば,  $n$  を  $n+1$  として 1. に戻る. ないならば, 終了する.

これには無駄が多い. もう少し標準的なアルゴリズムを考えよう. 素朴に考えて, ノード  $r$  に移動できるノードから順に更新していけば良さそうである. 以下では候補リストと書いているが, 純粋なリストと考えないように. キューを使うと良さそうである.

0.  $v_{j \neq r}^0 = M, v_r^0 = 0$ ,  $M$ : 十分大きな値, 候補リスト  $C = \{r\}$ ,  $n = 1$  とする.
1. 候補リストの左 (一番上) から, node  $j \in C$  を選ぶ.
2.  $j$  に移動できる node  $i \in I_j^-$  のすべてに対して,
  1.  $\hat{v}_i = c_{ij} + v_j$ .
  2.  $\hat{v}_i < v_i$  ならば let  $v_i = \hat{v}_i$  として,  $i \notin C$  ならば  $C$  の後ろに  $i$  を追加する.
3.  $j$  を  $C$  から取り除く.  $C$  が空でなければ, 1. に戻る. 空であれば終了する.

ほとんどすべての離散動的計画は最短経路問題と見做すことが出来る. 各ノード  $i$  がシステムの離散的な状態を表す. スタートノード  $q$  が初期状態であり, エンドノード  $r$  は任意の時刻  $T$  での任意の状態と見做せる.  $T \rightarrow \infty$  を考えることもできるが, その場合は Discount factor を含めることになるだろう.

### 2.1.2 The Discrete Budgeting Problem

$T$  個のタスクの集合  $\mathcal{T}$  に対する予算  $R$  の配分を考える. 各タスク  $t \in \mathcal{T}$  について, 割り当てる予算を  $a_t$ , 得られる報酬を  $C_t(a_t)$  とする. 次の量を最大化する  $a$  の組み合わせ  $\{a_t\}_{t \in \mathcal{T}}$  を求める.

$$\sum_{t \in \mathcal{T}} C_t(a_t) \quad (2.2)$$

但し, 予算に関する制約条件

$$\sum_{t \in \mathcal{T}} a_t = R \quad (2.3)$$

を満たす必要があり, さらに割当可能な予算は正

$$a_t \geq 0, \forall t \in \mathcal{T} \quad (2.4)$$

とする. これら 2 つの条件の下で報酬を最大化する問題を budgeting problem という.

$t = 1, \dots, T$  と順次タスクに予算を割り当てていくことを考える. 結局,

$R_t$   $t$  以降のタスクに割当可能な予算

$V_t(R_t)$  予算  $R_t$  を  $t$  以降のタスク  $t, t+1, \dots, T$  に割り当てた際の報酬総和の最大値 (以下では価値と呼ぶ).

という量を考え,  $V_t(R_t)$  を求める問題になる. 動的計画法の言葉では,  $R_t$  は状態変数,  $a_t$  はアクションと呼ばれる. 今の問題では,

$$R_{t+1} = R_t - a_t =: R^M(R_t, a_t) \quad (2.5)$$

となる。  $R^M$  は状態遷移を規定する関数であり、システムモデルと呼ばれる。次の最適性方程式 (optimality equation) が成立する。

$$V_t(R_t) = \max_{0 \leq a_t \leq R_t} \{C_t(a_t) + V_{t+1}(R^M(R_t, a_t))\} \quad (2.6)$$

最後のタスク  $T$  の次のタスク  $T+1$  は存在しないので、

$$V_T(R_T) = \max_{0 \leq a_T \leq R_T} C_T(a_T) \quad (2.7)$$

となる。これは便宜上  $V_{T+1}(R) = 0$  と当てはめて考えればよい。  $V_T(R_T)$  が求まれば  $t = T-1, T-2, \dots, 1$  の順に  $V_t(R_t)$  を求められることが分かる。解が存在する条件は、  $C_t(x)$  が有限であることのみであり、ここで説明した方法は budgeting problem の一般的な解法である。

計算量は、計算結果を適切に保持していれば、

$$\mathcal{O}(\text{size}(R)^2 \times T) :$$

となるはずである。

### 2.1.3 The Continuous Budgeting Problem

アクションが離散的な量の場合は  $a$ 、連続的な場合は  $x$  を用いることにする。連続量の budgeting problem を考える。特に、次の仮定を置いた場合の解析解について説明する：

$$C_t(x) = \sqrt{x}. \quad (2.8)$$

これは読めばすぐ分かるので省略する。

## 2.2 Stochastic Problems

### 2.2.1 Decision Trees

決定木 (けっていぎ)

### 2.2.2 A Stochastic Shortest Path Problem

経路のコストが確率的に定まる。どのタイミングで定まるかに応じて、問題が変わる。

次のノードに移動するコストが確率的な場合：

$$v_i^n = \min_{j \in I_i^+} \mathbb{E}\{c_{ij}(W) + v_j^{n-1}\}, W : \text{RandomVariable}. \quad (2.9)$$

次のノードに移動するコストは確定的であるが、それ以降は確率的である場合：

$$v_i^n = \mathbb{E} \left[ \min_{j \in I_i^+} \{c_{ij}(W) + v_j^{n-1}\} \right], W : \text{RandomVariable}. \quad (2.10)$$

確率過程と条件付き確率を用いて表現できそうだ。

### 2.2.3 The Gambling Problem

ギャンブラーが, 全  $N$  回の賭けを行う. 各回にどれだけの資金を賭けるべきか.

$p, q$  各回の勝ち負けの確率.  $q = 1 - p, q < p$ .

$S_n$   $n = 0, 1, \dots, N$  回目の賭け後の資産, この問題では状態と呼ぶ.

$a_n$   $n$  回目の賭け金額. 離散的な値とし,  $a_n \leq S_{n-1}$  という制約を課す.

$W_n$   $n$  回目の賭けにギャンブラーが勝つ場合, 1 負ける場合 0 となる確率変数.

$V_n(S_n)$   $n$  回目の賭け後に  $S_n$  の資産を保持していることの価値. この価値は, 将来的に得られる価値の期待値とするもので, 最終時点  $N$  での価値については  $\ln(S_N)$  と定義する.

$V_0(S_0)$  を最大化するような  $\{a_n\}_{n \geq 0}$  を求めたい. これは金融工学で出てくる, デリバティブ価格の評価の話と同一である.

$S_n, V_n$  に関する漸化式は次のようになる:

$$S_{n+1} = S_n + a_{n+1}W_{n+1} - (1 - W_{n+1})a_{n+1} =: R^M(S_n, a_{n+1}), \quad (2.11)$$

$$V_n(S_n) = \max_{0 \leq a_{n+1} \leq S_n} E[V_{n+1}(R^M(S_n, a_{n+1})) | S_n] \quad (2.12)$$

$$= \max_{0 \leq a_{n+1} \leq S_n} E[V_{n+1}(S_n + a_{n+1}W_{n+1} - (1 - W_{n+1})a_{n+1}) | S_n] \quad (2.13)$$

価値  $V_n(S_n)$  は,  $n$  回目の賭けが終了した時点での情報に基づいて,  $n + 1$  回目以降の賭けを確率的に生じるものと見做した期待値を最大化するものとして表現している.

$n$  の大きい値から 0 に向けて考える.  $N$  のとき  $V_N(S_N) = \ln S_N$  である.  $N - 1$  のとき,

$$V_{N-1}(S_{N-1}) = \max_{0 \leq a_N \leq S_{N-1}} E[\ln(S_{N-1} + a_N W_N - (1 - W_N)a_N) | S_{N-1}] \quad (2.14)$$

$$= \max_{0 \leq a_N \leq S_{N-1}} [p \ln(S_{N-1} + a_N) + (1 - p) \ln(S_{N-1} - a_N)] \quad (2.15)$$

$\max$  内の関数は  $a_N$  について凸なので, 極値を調べることで  $\max$  を与える  $a_N$  と解析解を求めることができる. 結果,  $a_n$  と  $V_n$  が各  $n$  で同様の関数型となることと最適解が  $p > 0.5$  を前提とすることが分かる.

### 2.2.4 Asset Valuation

### 2.2.5 The Asset Acquisition Problem I

時点  $t$  にインターバル  $t + 1$  に使用される分の products を  $x$  購入する.  $x$  は一般には連続量であり, 多次元ベクトルとなりうる.

$R_t$  assets on hand at time  $t$  before we make a new order decision, and before we have satisfied any demands arising in time interval  $t$ ,

$x_t$  amount of products purchased at time  $t$  to be used during time interval  $t + 1$ ,



$\hat{D}_t$  random demands that arise between  $t - 1$  and  $t$ ,

$p^p, p^s$  固定の purchase 価格, sell 価格,

$C_t(x_t)$  the amount we earn between  $t - 1$  and  $t$ , including the decision we make at time  $t$ .

$$R_{t+1} = R_t - \min(R_t, \hat{D}_t) + x_t \quad (2.16)$$

$$C_t(x_t) = p^s \min(R_t, \hat{D}_t) - p^p x_t \quad (2.17)$$

period  $t$  で要求される  $\hat{D}_t$  に応える (売り出す) 前の量として  $R_t$  を定義していることに注意する. 後のセクションのためにこのように定義しておくとのこと. period と interval という複数のおそらく同一の意味の言葉が出てきていたり, その定義が明確でないので分かりにくい.

区間  $(t - 1, t]$  をインターバル  $t$  と呼び,  $R_t$  は  $t - 1$  直後の時点での在庫量,  $\hat{D}_t$  はインターバル  $t$  内, 厳密に考えると  $(t - 1, t)$  で発生する需要の総和で,  $x_t$  は  $\hat{D}_t$  を満たした直後の時点  $t$  に購入する量ということになるだろう. 在庫量の変化の流れを  $R_t$  を用いて表すと, 次のように考えていることになる.

$$R_t \rightarrow (R_t - \hat{D}_t) \rightarrow (R_t - \hat{D}_t + x_t) = R_{t+1}$$

$C_t(x_t)$  の定義も腑に落ちる.

Contribution の総和を最大化するという問題については, ベルマン方程式によって解くことが出来る.  $R_t$  が状態変数であり,  $V_t(R_t)$  を価値関数と定義して, これを最大化する.

$$V_t(R_t) = \max_{x_t} (C_t(x_t) + \gamma E\{V_{t+1}(R_{t+1})\}) \quad (2.18)$$

これは,  $t$  時点までの情報を  $\mathcal{F}_t$  として条件付き確率を用いつつ,  $\hat{D}(t)$  も明示して書くと,

$$V_t(R_t, \hat{D}_t | \mathcal{F}_t) = \max_{x_t} (C_t(x_t, \hat{D}_t) + \gamma E\{V_{t+1}(R_{t+1}, \hat{D}_{t+1} | \mathcal{F}_t)\}) \quad (2.19)$$

となり,  $\hat{D}_{t+1}, \dots$  を通して確率的な量の期待値を考えることになることが分かる.

### 2.2.6 The Asset Acquisition Problem II

多くの asset 取得問題においては需要の他にも不確実性の原因となるものがある. まずは価格である. さらに, 保持している量に対する外生的な増減がありうる.

需要と供給に関連する記号を定義する.

$x_t^p$  asset purchased(acquired) at time  $t$  to be used during time interval  $t + 1$ ,

$x_t^s$  amount of assets sold to satisfy demands during time interval  $t$ ,

$x_t$   $(x_t^p, x_t^s)$ ,

$R_t$  resource level at time  $t$  before any decisions are made,

$D_t$  demands waiting to be served at time  $t$ .

在庫が無かったり需要が無ければ売れない, ということで  $x_t^s \leq \min(R_t, D_t)$  としておく.

$D_t$  の定義をみると,  $t$  まで serve するのを待たされている需要, とある. 前の節でみたように,  $(t-1, t)$  に発生する需要の総和であり, それらは  $t$  丁度に満たされると解釈しようということかと思っただが, 需要が累積することを言いたいのかもしれない.

次は価格である.

$p_t^p$  market price for purchasing assets at time  $t$ ,  
 $p_t^s$  market price for selling assets at time  $t$ ,  
 $p_t$   $(p_t^p, p_t^s)$ .

そして外生的な変動を表す量.

$\hat{R}_t$  exogenous changes to the asset levels on hand that occur during time interval  $t$ ,  
 $\hat{D}_t$  demand for the resources during time interval  $t$ ,  
 $\hat{p}_t^p$  change in the purchase price that occurs between  $t-1$  and  $t$ ,  
 $\hat{p}_t^s$  change in the selling price that occurs between  $t-1$  and  $t$ ,  
 $\hat{p}_t$   $(\hat{p}_t^p, \hat{p}_t^s)$ .

価格そのものを直接的に確率変数とするのではなく, 変化分を確率的なものとして定式化している. 需要についてもどうも増加分を確率的なものとして定義していると解釈するのが適切にみえる.

これらはまとめて扱うと見通しが良い. 外生的な変動を  $W_t$ , 状態を  $S_t$  とする;

$$W_t = (\hat{R}_t, \hat{D}_t, \hat{p}_t) \quad (2.20)$$

$$S_t = (R_t, D_t, p_t) \quad (2.21)$$

状態変数の transition を generically に表す. システムモデルと呼んだり, 遷移関数 (transition function) と呼んだりするものである.

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \quad (2.22)$$

具体的な遷移関数の形を書いてみよう.

$$R_{t+1} = R_t - x_t^s + x_t^p + \hat{R}_t, \quad (2.23)$$

$$D_{t+1} = D_t - x_t^s + \hat{D}_{t+1}, \quad (2.24)$$

$$p_{t+1}^p = p_t^p + \hat{p}_t^p, \quad (2.25)$$

$$p_{t+1}^s = p_t^s + \hat{p}_t^s. \quad (2.26)$$

1 期間の貢献関数

$$C_t(S_t, x_t) = p_t^s x_t^s - p_t^p x_t^p. \quad (2.27)$$

最適意思決定はベルマン方程式を解けば求まる:

$$V_t(S_t) = \max_{x_t} (C_t(S_t, x_t) + \gamma E\{V_{t+1}(S^M(S_t, x_t, W_{t+1}) \mid \mathcal{F}_t)\}) \quad (2.28)$$

### 2.2.7 The Lagged Asset Acquisition Problem

将来の各時点での需要に対して前もって asset を買うことを考えよう. 具体例としては, 旅行代理店がホテルを先に予約購入したり, 航空券を買うような場合がある.

$x_{t,t'}$  assets purchased at time  $t$  to be used to satisfy demands that become known during time interval between  $t' - 1$  and  $t'$ ,

$x_t$   $(x_{t,t+1}, x_{t,t+2}, \dots) =: (x_{t,t'})_{t' \geq t}$ ,

$\hat{D}_t$  demands for the resources that become known during time interval  $t$ ,

$R_{t,t'}$  total assets acquired on or before time  $t$  that may be used to satisfy demands that become known between  $t' - 1$  and  $t'$ ,

$R_t$   $(R_{t,t'})_{t' \geq t}$ .

$\hat{D}_t$  の需要に対して  $x_{t,t}$  を反映させることはできないものとする. 在庫は前もって必要ということである. 出荷に時間がかかると考えてもいいだろう. 制約  $x_{t,t} = 0$  を課すということである.

遷移関数は

$$R_{t+1,t'} = \begin{cases} (R_{t,t} - \min(R_{t,t}, \hat{D}_t)) + x_{t,t+1} + R_{t,t+1}, & t' = t + 1 \\ R_{t,t'} + x_{t,t'}, & t' \geq t \end{cases} \quad (2.29)$$

貢献関数は, 価格を  $p^p, p^s$  も定義されているとして,

$$C_t(R_t, \hat{D}_t) = p^s \min(R_{t,t}, \hat{D}_t) - \sum_{t' > t} p^p x_{t,t'} \quad (2.30)$$

最適行動 (価値を最大化する  $x_{t,t'}$  の決定) については, やはりベルマン方程式を解けばよい. が, 本節の  $x_t$  や  $R_t$  は多次元ベクトルであり, すべての状態と行動を数値的に列挙するのは事実上不可能であるところが, 前節までと異なる.

### 2.2.8 The Batch Replenishment Problem

### 2.2.9 The Transformer Replacement Problem

### 2.2.10 The Dynamic Assignment Problem

## 2.3 Information Acquisition Problems

### 2.3.1 The Bandit Problem

ギャンブラーが  $K$  個のスロットマシンのうち, どれを選ぶかを考えている. 各スロットマシンの勝率はそれぞれ固定された値であるとするが, どれがどの値かはギャンブラーは知らない. プレイすることで勝率に関するデータを収集することができる.

$$x_k^n = \begin{cases} 1 & \text{if we choose to play the } k\text{th slot machine in the } n\text{th trial,} \\ 0 & \text{otherwise.} \end{cases}$$

$W_k^n$  winnings from the  $k$ th slot machine during the  $n$ th trial. 勝ちの場合の儲け額

$\bar{w}_k^n$  our estimate of the expected winnings from playing the  $k$ th slot machine after the  $n$ th trial.

$(\bar{s}_k^2)^n$  our estimate of the variance of the winnings from playing the  $k$ th slot machine after the  $n$ th trial.

右肩の  $n$  はべき乗ではなく  $n$  番目を表すもの.

$N_k^n$  number of times after we have played the  $k$ th slot machine after  $n$ th trials.

素朴にはこの winnings の平均が大きく, 分散が小さいスロットを使うのが良さそうに見える. 無限回試行できるなら平均だけに着目すればよいだろう.

$x_k^n = 1$  ならば  $W_k^n$  は何らかの実現値となっているので, 次の式で推定値を更新できる:

$$\bar{w}_k^n = \left(1 - \frac{1}{N_k^n}\right) \bar{w}_k^{n-1} + \frac{1}{N_k^n} W_k^n, \quad (2.31)$$

$$(\bar{s}_k^2)^n = \frac{N_k^n - 2}{N_k^n - 1} (\bar{s}_k^2)^{n-1} + \frac{1}{N_k^n} (W_k^n - \bar{w}_k^{n-1})^2. \quad (2.32)$$

これらは  $\mathcal{N}_k^n = \{m : x_k^m = 1, m = 0, 1, \dots, n\}$  として定義できる自然な平均と分散である:

$$\begin{aligned} \bar{w}_k^n &= \frac{1}{N_k^n} \sum_{m \in \mathcal{N}_k^n} W_k^m, \\ (\bar{s}_k^2)^n &= \frac{1}{N_k^n - 1} \sum_{m \in \mathcal{N}_k^n} (W_k^m - \bar{w}_k^m)^2 \\ &= \frac{1}{N_k^n - 1} \sum_{m \in \mathcal{N}_k^{n-1}} \{(W_k^m - \bar{w}_k^m)^2\} + \frac{1}{N_k^n - 1} (W_k^n - \bar{w}_k^n)^2 \\ &= \frac{N_k^n - 2}{N_k^n - 1} (\bar{s}_k^2)^{n-1} + \frac{1}{N_k^n - 1} \left( W_k^n - \frac{1}{N_k^n} W_k^n - \left( \frac{N_k^n - 1}{N_k^n} \right) \bar{w}_k^{n-1} \right)^2 \\ &= \frac{N_k^n - 2}{N_k^n - 1} (\bar{s}_k^2)^{n-1} + \frac{1}{N_k^n} (W_k^n - \bar{w}_k^{n-1})^2. \end{aligned}$$

$\bar{w}_k^n, (\bar{s}_k^2)^n$  を状態変数として採用することで,  $n$  回目の試行の後のシステムの状態  $S^n$  を次のように定義できる:

$$S_n = (\bar{w}_k^n, (\bar{s}_k^2)^n, N_k^n)_{k=1,2,\dots,K}$$

このように定義しておくことで,  $S_n$  から  $S_{n+1}$  を求めることができる.  $S_n$  の要素は  $N_k^n$  分を除くと  $2K$  個あり, それぞれ一般に連続変数である.

$S_n$  は”知識の状態”, つまりこの系に関する知識の程度を表すものと考えることができる.  $\bar{w}_k^n$  は文字通り平均的な winnings を表しており,  $(\bar{s}_k^2)^n$  は勝率についてどの程度正確に知っているかを表している.  $S_n$  は”information state”もしくは”hyperstate”と呼ばれることがある.

これは純粋な information acquisition problem となっている。推定期待値最大となる action をとる, すなわち  $\bar{w}_k^n$  が最大となるスロットを選んでプレイするのが良いと思えるが, これは正しくない。推定した期待値が正しくない場合に誤った戦略をとることになるためである。推定期待値が最大でないスロットを選ぶことも必要となる。

理論的にはベルマン方程式により解くことが出来る。

$$V_n(S_n) = \max_{x^n=(x_1^n, \dots, x_K^n), \sum x_k^n=1} \left( \sum_{k=1}^K x_k^{n+1} W_k^{n+1} + V_{n+1}(S_{n+1}) \mid S_n \right) \quad (2.33)$$

この方程式は状態変数  $S_n$  が連続変数であること,  $W_k^n$  の分布も分かっていない<sup>\*1</sup> ことから compute するのは困難である。しかしながら,  $S_n$  に含まれる推定された平均と分散により, 真の分布に関する各スロットの平均と分散について infer できる。

この bandit problem の他にも多数の information acquisition problems があるので以下, いくつか紹介しておく。また bandit problem を含む幾つかの問題は index policy という方法を用いることで解くことが出来る。K 次元の問題を K 個の 1 次元の問題に帰着できるらしいが, これは後述となる。

Example2.3 引っ越した後, 自宅から職場までの時間に関する平均最短経路を見つけたい。

Example2.4 野球チームの監督が, 複数の野球選手のうち, ヒットを良く打つ選手を選びたい。

Example2.5 資産管理を資産マネジャーに任せたい。誰に任せるといいか。どの時点で切り替えるといいか。

Example2.6 A doctor is trying to determine the best blood pressure medication for a patient.

Each patient responds differently to each medication, so it is necessary to try a particular medication for a while, and then switch if the doctor feels that better results can be achieved with a different medication.

前二つは統計とれば済むという気がする。後ろ二つは本質的に難しい話に見える。これは自分が感じる結果の軽重度合いの問題かもしれない。

### 2.3.2 An Information-Collecting Shortest Path Problem

各リンクの平均・分散コストも分からないような Stochastic Shortest Path Problem を考える。bandit problem との違いに着目する。

ノード  $i$  から選べる次のノード  $j_1, j_2$  から  $j_1$  を選択して得られる情報 (コスト) は,  $j_2$  を選んだ場合の  $j_2$  から Goal までのコストに関する情報を含みうる。各ノードが共通の経路を含みうるためである。bandit problem では, 各スロットの結果はそれぞれのスロットに関する情報にしかならなかった。

また,  $j_1$  を選んだことにより次に選べるノードは  $j_2$  から選べるものと異なる。bandit problem では, どのスロットを選んだ後であっても任意のスロットを選択できる。

simple bandit problems については解けているが, このような集めた情報と将来の選択肢に影響を

<sup>\*1</sup> そもそも  $W_k^n$  の分布が分かっていたら, 個別に期待値を計算すれば良いだけである。

与えうる問題については, 未だ解かれていない.

## 2.4 A Simple Modeling Framework for Dynamic Programs

Stochastic Dynamic Programs を構成する要素

**State Variable.** This captures all the information we need to make a decision, as well as the information that we need to describe how the system evolves over time.

**Decision Variable.** Decisions/actions we represent how we control the process.

**Exogenous information.** This is data that first become known each time period (e.g., the demand for product, or the price at which it can be purchased or sold). In addition we have to be told the initial state of our system.

**Transition function.** This function determines how the system evolves from the state  $S_t$  to the State  $S_{t+1}$  given the decision that was made at time  $t$  and the new information arrived between  $t$  and  $t + 1$ .

**Objective function.** This function specifies the costs being minimized, or contributions/rewards being maximized, over a time horizon.

2.2.6 節を用いて説明している.

Contribution function  $C_t(S_t, a_t)$  について, 一般には  $C_t(S_t, a_t, W_{t+1})$  となる場合もある, ということが触れられていたり, decision function  $A^\pi$  の記号が導入されている. それくらい.

## 第 3 章

# Introduction to Markov Decision Processes

状態空間と行動空間 (取りうる action の集合) が小さい場合を扱う.

### 3.1 The Optimality Equations

次のような目的関数の求解を含む問題を前章で多く見た:

$$\max_{\pi} E^{\pi} \left( \sum_{t=0}^T \gamma^t C_t^{\pi}(S_t, A_t^{\pi}(S_t)) \right) \quad (3.1)$$

$E^{\pi}$  の  $\pi$  が付いたり付かなかったりするが, これは選択した行動が取得可能な情報に影響を与えるか与えないかによって, 変わる. 詳しくは 5 章で説明するらしい.

通常, この求解問題を数値的に取り扱うことは困難であるが, いろいろ示唆を導くことが出来る.

#### 3.1.1 Bellman's Equations

deterministic shortest path problem を考える. 状態  $S_t$  の取りうる値をパス上のノードのインデックス  $i$ , ノード  $j$  へ移るアクション  $a_t$  をそのノードのインデックス  $j$  で表すことにする.

$$\begin{aligned} S_t &= i \\ a_t &= j \end{aligned}$$

さらに遷移関数  $S^M(S_t, a_t)$  <sup>\*1</sup> を導入しておく.  $S_{t+1} = S^M(S_t, a_t)$  であり,  $S_{t+1}$  は  $S_t$  と  $a_t$  で確定的に定まる.

$t + 1$  での状態  $S_{t+1}$  の価値  $V_{t+1}(S_{t+1})$  が与えられていると仮定しよう. 1 期間の貢献関数 (contribution function)  $C_t(S_t, a_t)$  と 1 期間分だけある  $\gamma$  で  $V_{t+1}(S_{t+1})$  を割り引いた値の和を最大化する問題

$$a_t^*(S_t) = \arg \max_{a \in \mathcal{A}_t} (C_t(S_t, a) + \gamma V_{t+1}(S^M(S_t, a))) \quad (3.2)$$

---

<sup>\*1</sup> 実はこれは上のアクションの定義から分かるように  $a_t$  を返す関数である.

を解きたい.  $\mathcal{A}_t$  は時刻  $t$  の状態が  $S_t$  のときの取りうるアクションの集合であって,  $\mathcal{A}_t(S_t)$  とでもかくべきものである.

$S_t$  での価値は  $a_t^*(S_t)$  を選んだときの価値であり,

$$V_t(S_t) = C_t(S_t, a_t^*(S_t)) + \gamma V_{t+1}(S^M(S_t, a_t^*(S_t))) \quad (3.3)$$

となる.

これは状態・行動ペアの価値関数  $V_t(s, a)$

$$V_t(s, a) = C_t(s, a) + \gamma V_{t+1}(S^M(s, a))$$

を導入すると見通しが良いのではないか. 解くべき問題を

$$V_t(S_t) = \max_{a \in \mathcal{A}_t(S_t)} V_t(S_t, a)$$

と書き直すことが出来る.  $V_t(S_t) = V_t(S_t, a_t^*(S_t))$  である.

次に, stochastic problems を考えよう. 行動を選択した後に新たな情報が入ってくるものとする. より具体的には,  $C_t(S_t, a_t)$ ,  $S^M(S_t, a_t)$  が確定的でなくなり, 行動選択後の何らかの情報を加えて定まるものとする.

石油精製所の在庫の時間発展を考えよう.

$S_t$  時点  $t$  の石油在庫量.

$a_t$   $t$  に order され,  $t$  と  $t+1$  の間に利用可能になる石油量. 生成する量と考えれば良い.

$\hat{D}_{t+1}$   $t$  と  $t+1$  の間の石油需要量. 出荷する量と考えれば良い.

時間順序としては  $S_t \rightarrow a_t \rightarrow \hat{D}_{t+1}$  という感じか.

$S_{t+1}$  は次のように書ける. これは遷移関数でもある.

$$S_{t+1}(S_t, a_t, \hat{D}_{t+1}) = \max(0, S_t + a_t - \hat{D}_{t+1}).$$

$\hat{D}_{t+1}$  の確率分布<sup>\*2</sup> を  $P^D(x) = \text{Prob}(\hat{D} = x)$  とすると,  $S_{t+1}$  の確率分布は

$$\text{Prob}(S_{t+1} = s') = \begin{cases} 0, & s' > S_t + a_t \\ \text{Prob}(S_t + a_t - \hat{D}_{t+1} = s') = P^D(S_t + a_t - s'), & 0 < s' \leq S_t + a_t \\ \sum_{d=S_t+a_t}^{\infty} P^D(d), & s' = 0. \end{cases}$$

この確率分布が  $S_t, a_t$  に依存していることが分かるので,

$$P(S_{t+1} | S_t, a_t)$$

という条件付き確率で表す. これを用いると解くべき問題は

$$V_t(S_t) = \max_{a \in \mathcal{A}_t} \left( C_t(S_t, a) + \gamma \sum_{s' \in \mathcal{S}} P(S_{t+1} = s' | S_t, a_t) V_{t+1}(s') \right) \quad (3.4)$$

<sup>\*2</sup> ここでは probability mass function である.



と書ける. この形はよく使われているので, 本書ではベルマン方程式の標準形と呼ぶことにする. ADP 向けの等価な表現は期待値記号を用いたもので,

$$V_t(S_t) = \max_{a \in \mathcal{A}_t} (C_t(S_t, a) + \gamma E[V_{t+1}(S_{t+1}(S_t, a_t, W_{t+1})) | S_t]). \quad (3.5)$$

これをベルマン方程式の期待値形と呼ぶことにする. この形が後の章でのアルゴリズム関連の議論の基礎となる.

ところで  $S_{t+1}$  の  $S_t, a_t$  依存性については, 関数型としての依存性と確率的な意味での依存性の 2 通りが考えられるので注意するようにと書かれている. しかし, あまり気にする必要を感じない. どの部分が確率的か分かっていれば特に問題無いだろう.

標準形がよく使われているのは, 行列形式で問題を記述できるかららしい.

$$p_{ss'}(a) := P(S_{t+1} = s' | S_t = s, a_t = a)$$

これは  $s$  から  $a$  を選択したときの  $s'$  への遷移確率を表しており,  $ss'$  が行列の要素を表すと見做せる.

状態  $S_t = s$  に対してアクションを返す関数の集合を考える. その要素  $A^\pi$  は  $A^\pi(S_t) \in \mathcal{A}_t$  を満たす. また  $A_t^\pi(s) := A^\pi(S_t = s)$  をとしておく.<sup>\*3</sup> 各  $A^\pi$  は各状態でどのような行動を取るかを定めるものであり, これをポリシーと呼ぶことにする. ベルマン方程式を解くという問題は, 価値を最大化するポリシー  $A^\pi$  を探すという問題と言える.

ポリシー  $A^\pi$  に関する遷移確率は

$$p_{ss'}^\pi := P(S_{t+1} = s' | S_t = s, a_t = A_t^\pi(s))$$

となる. これらを要素とする行列を  $P_t^\pi$  と書くことにする.

この辺りはより丁寧に,

$$P_t^\pi = (p_{t,ss'}^\pi)$$

$$p_{t,ss'}^\pi := p_{t,ss'}(A_t^\pi(s)) := P(S_{t+1} = s' | S_t = s, a_t = A_t^\pi(s))$$

とでも書くべきな気はする.

記号を整理する:

$v_t$   $V_t(s)$  を要素として持つ  $s$  に関する縦ベクトル.

$c_t^\pi$   $C_t(s, A_t^\pi(s))$  を要素として持つ  $s$  に関する縦ベクトル.

$P_t^\pi$   $\pi$  で指定されるポリシー (略してポリシー  $\pi$ ) に対する時点  $t$  での 1 step 遷移確率行列.

ベルマン方程式は次のように書ける.

$$v_t = \max_{\pi} (c_t^\pi + \gamma P_t^\pi v_{t+1}) \quad (3.6)$$

<sup>\*3</sup> これは  $A^\pi$  が関数としては  $t$  依存性を持たないことを意図している. もしかすると, 陽に  $t$  依存性を持つものとしてポリシーというものを定義するべきかもしれないが, 取りあえずこうしておく. 違ったら読み替えてほしい.



## 第 4 章

# Dynamic Resource Allocation Problems

### 4.1 An Asset Acquisition Problem