2017. 10. 20 (Fri) デザインシステム学 (槌木，中西)

入学年度	学年	研究室名	氏名
2017	M1	槌木	梅杰 (MEI JIE)

1.

次の問題を定式化し，答えを求めよ.

1) 各辺の長さの和が p である長方形の中でその面積が最大であるもの.

2) 面積が a である長方形の中で 4 辺の長さの和が最小となるもの.

1) set x is the length and y is the width

$2(x+y)=p$

since $s=x\cdot y$, so $s=x\cdot(\frac{p}{2}-x)=-x^2+\frac{p}{2}x$

according to the principle that when $s'=0$, s get maximum or minimum

$s'=\frac{p}{2}-2x=0 \Rightarrow x=\frac{p}{4}$

so when $x=\frac{p}{4}$ $y=\frac{p}{4}$, $S_{max}=\frac{p^2}{16}$

2) set x is the length and y is the width

$a=x\cdot y$

since $p=2(x+y)$, so $p=2(x+\frac{a}{x})=\frac{2x^2+2a}{x}$

according to the principle that when $p'=0$, p get maximum or minimum

$p'=\frac{4x\cdot x-(2x^2+2a)}{x^2}=0 \Rightarrow x=\sqrt{a}$

so when $x=\sqrt{a}$ $y=\sqrt{a}$, $P_{min}=4\sqrt{a}$

(鞍点)

2. (驻点)

スカラー値評価関数 $L(x,u)$ の停留点を求めよ．その停留点は最大点，最小点，按点のいずれであるかを調べよ．

$$L(x,u)=\frac{1}{2}x^2+xu+u^2+u$$

次の拘束条件 $f(x,u)=0$ が付加されたとき，スカラー値評価関数 $L(x,u)$ の停留点を求めよ．その停留点は最大点，最小点，按点のいずれであるかを調べよ．

$$f(x,u)=x-3=0$$

1) according to the function above, we can get

$$\frac{\partial L}{\partial x}=x+u \qquad \frac{\partial L}{\partial u}=2u+(x+1)$$

when $\frac{\partial L}{\partial x}=\frac{\partial L}{\partial u}=0$, $L(x,u)$ get maximum or minimum

$$\begin{cases} x+u=0 \\ 2u+x+1=0 \end{cases} \Rightarrow \begin{cases} x=1 \\ u=-1 \end{cases} \Rightarrow (x,u)=(1,-1)$$

2) let $H=L+\lambda f=\frac{1}{2}x^2+xu+u^2+u+\lambda(x-3)$, so we can get

$$\frac{\partial H}{\partial x}=x+u+\lambda \qquad \frac{\partial H}{\partial u}=x+2u+1 \qquad \frac{\partial H}{\partial \lambda}=x-3$$

when $\frac{\partial H}{\partial x}=\frac{\partial H}{\partial u}=\frac{\partial H}{\partial \lambda}=0$, H get maximum or minimum

$$\begin{cases} x+u+\lambda=0 \\ x+2u+1=0 \\ x-3=0 \end{cases} \Rightarrow \begin{cases} x=3 \\ u=-2 \\ \lambda=-1 \end{cases} \Rightarrow (x,u)=(3,-2)$$

$$L(x,u)=\frac{1}{2}x^2+xu+u^2+u$$

$$\Rightarrow \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + [0.\ 1] \begin{bmatrix} x \\ u \end{bmatrix}$$

$$\Downarrow \quad \Downarrow \quad \Downarrow \quad \Downarrow \quad \Downarrow$$

$$\Rightarrow \mathbb{X}^T \quad \mathbb{A} \quad \mathbb{X} + \mathbb{b}^T \quad \mathbb{X}$$

$$\Rightarrow \frac{1}{2}\mathbb{A}\mathbb{u}+\mathbb{b}^T=0$$

(calculation with matrix).

入学年度	学年	研究室名	氏名
29	M1	椹木	梅杰 (MEI JIE)

システムの方程式を

$$x_{k+1} = x_k + u_k, \quad k = 0, 1, 2, 3, 4 \tag{1}$$

評価関数を

$$S = \sum_{k=0}^{3} |x_k| + |u_k| \tag{2}$$

境界条件を

$$x_0 = 5, x_4 = 0 \tag{3}$$

とするときの最適制御入力 $u_k^*, k = 0, 1, 2, 3$ を求めよ．ただし，u_k は整数とする．

According to the Principle of Optimality from Richard Bellman, an optimal policy has the property that whatever the initial state and intial decision are, the remaining decision must constitute an optimal policy with regard to the state resulting from the first decision. In other words, an optimal problem can be devided into several sub-optimal problem and backward propagated along the sub-problem to solve a global optimal problem.

⇒ calculation with backward propagation.

① $S_4^* = S_3^* + R_4^* = S_3^* + [|x_4| + |u_4|]^* = S_3^* + [|x_3 + u_3|]^* + |u_4|$

⇒ when $u_4 = 0$, $S_4^* = S_3^* + [|x_3 + u_3|]^*$

② $S_4^* = S_2^* + [|x_3| + |u_3|]^* + [|x_3 + u_3|]^*$

⇒ when $u_3 = -x_3$, $S_4^* = S_2^* + [2|x_3|]^*$

③ $S_4^* = S_1^* + [|x_2| + |u_2|]^* + [2|x_2 + u_2|]^*$

⇒ when $u_2 = -x_2$, $S_4^* = S_1^* + [2|x_2|]^*$

④ $S_4^* = S_0^* + [|x_1| + |u_1|]^* + [2|x_1 + u_1|]^*$

⇒ when $u_1 = -x_1$, $S_4^* = S_0^* + [2|x_1|]^*$

⑤ $S_4^* = [|x_0| + |u_0|]^* + [2|x_0 + u_0|]^* = 5 + [|u_0|]^* + 2[|5 + u_0|]^*$

⇒ when $u_0 = -x_0$, $S_4^* = 5 + |-5| + |0| = 10$

conculsion:

$u_0 = -x_0 = -5$

$u_1 = -x_1 = 0$

$u_2 = -x_2 = 0$

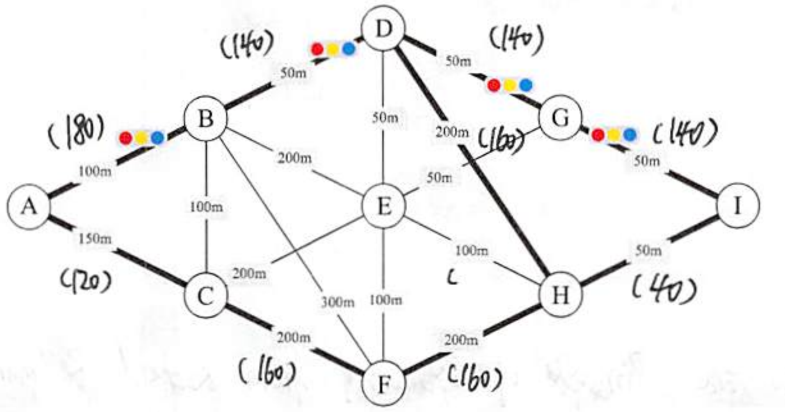
$u_3 = -x_3 = 0$

$u_4 = -x_4 = 0$

$S_4^* \Rightarrow S_3^* = 10$.

図1のような道路図を考える．図中のA～Iは交差点を表し，交差点間の線分は道路を表し，付されている数字は交差点間距離である．線分の太さは道路の車線数の大小を表し，細い線分は車線数が少なく，太い線分は車線数が多い．車線数が多いほど，単位移動距離に必要なコストが小さい．信号機マーク付された道路には，信号機が存在し，通過するには待ち時間のコストが必要となる．単位距離当たりの移動コスト，信号通過コストを適当に定め，A から I に移動するときの最小コスト経路を求めよ．また，最短距離ルートも求め，最小コスト経路を図中に示して比較せよ．

例：単位距離当たりの移動コスト（太い道路：0.8，細い道路：1.0），信号通過コスト（100）



(~) ⇒ cost
~ ⇒ distance.

図1　road map

from \ to	A	B	C	D	E	F	G	H	I
A		(180) 100	(120) 150	∞	∞	∞	∞	∞	∞
B			100	(140) 50	200	300	∞	∞	∞
C				∞	200	(160) 200	∞	∞	∞
D					50	∞	(140) 50	(160) 200	∞
E						100	50	100	∞
F							∞	(160) 200	∞
G								∞	(140) 50
H									(40) 50
I									

② considering the optimal route with distance

(same as ① and $D[I] = 250$)

⇒ optimal route is $A \to B \to D \to G \to I$

① considering the optimal route with cost.

⇒

	B	C	D	E	F	G	H	I
A	180	120	∞	∞	∞	∞	∞	∞

$[AC] = 120 < [AB] = 180 \Rightarrow$ pick $[AC]$

$D[C] = 120$ updates in accord with follow:

$$\begin{cases} D[B] = 180 < D[C] + [CB] = 220 \\ D[E] = \infty > D[C] + [CE] = 320 \\ D[F] = \infty > D[F] + [CF] = 280 \end{cases}$$

⇒

	B	C	D	E	F	G	H	I
A	180	120	∞	320	280	∞	∞	∞

$[AF] = 280 < [AE] = 320 \Rightarrow$ pick $[AF]$

$D[F] = 280$ updates in accord with follow:

$\{ D[H] = \infty > D[F] + [FH] = 440$

⇒

	B	C	D	E	F	G	H	I
A	180	120	∞	320	280	∞	440	∞

$D[I] = \infty > D[H] + [HI] = 480$

⇒ optimal route is $A \to C \to F \to H \to I$

入学年度	学年	研究室名	氏名
2017	M1	椹木研	梅杰 (MEI JIE)

3つ以上の行列の積を計算する場合，順番を変えても最終的な結果は変わらない．つまり，

$$A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

である．行列 A_1，A_2 をそれぞれ $(a_0 \times a_1)$，$(a_1 \times a_2)$ の実行列とすると，$A_1 \cdot A_2$ を求めるためには $(a_0 \cdot a_1 \cdot a_2)$ 回の掛算と和算がそれぞれ必要である．

4個の行列 $A_i (i = 1, 2, 3, 4)$ の積 $A_1 \cdot A_2 \cdot A_3 \cdot A_4$ をもっとも効率よく計算できるように括弧をつけて行列積を表しなさい．ただし，各行列の型は以下の表の通りである．

A_1	A_2	A_3	A_4
30×35	35×15	15×5	5×10

$$\begin{cases} a_0 = 30 \\ a_1 = 35 \\ a_2 = 15 \\ a_3 = 5 \\ a_4 = 10 \end{cases}$$

complexity calculation :

① $A_1 \to A_2 \Rightarrow a_0 \cdot a_1 \cdot a_2 = 15750$

$A_2 \to A_3 \Rightarrow a_1 \cdot a_2 \cdot a_3 = 2625$

$A_3 \to A_4 \Rightarrow a_2 \cdot a_3 \cdot a_4 = 750$

② $A_1 \to A_3 \Rightarrow \begin{cases} (A_1 \cdot A_2) \cdot A_3 \Rightarrow (a_0 a_1 a_2) + (a_0 a_2 a_3) = 18000 \\ A_1 \cdot (A_2 \cdot A_3) \Rightarrow (a_0 a_2 a_3) + (a_1 a_2 a_3) = 7875 \Rightarrow \text{optimal} \end{cases}$

$A_2 \to A_4 \Rightarrow \begin{cases} (A_2 \cdot A_3) \cdot A_4 \Rightarrow (a_1 \cdot a_2 \cdot a_3) + (a_1 \cdot a_3 \cdot a_4) = 4375 \Rightarrow \text{optimal.} \\ A_2 \cdot (A_3 \cdot A_4) \Rightarrow (a_1 \cdot a_2 \cdot a_4) + (a_2 \cdot a_3 \cdot a_4) = 6000 \end{cases}$

③ $A_1 \to A_4 \Rightarrow \begin{cases} (A_1 \cdot A_2 \cdot A_3) \cdot A_4 \Rightarrow A_1 \cdot (A_2 \cdot A_3) \cdot A_4 = 9375 \Rightarrow \text{optimal} \\ (A_1 \cdot A_2) \cdot (A_3 \cdot A_4) \Rightarrow 21000 \\ A_1 \cdot (A_2 \cdot A_3 \cdot A_4) \Rightarrow A_1 \cdot (A_2 \cdot A_3) \cdot A_4 \Rightarrow \text{not necessary} \\ A_1 \cdot (A_2 \cdot A_3) \cdot A_4 \Rightarrow \text{not necessary} \end{cases}$

In short, the optimum is $A_1 \cdot (A_2 \cdot A_3) \cdot A_4$ and the calculational complexity is 9375

2017.12.22(Fri) デザインシステム学 (椹木，中西)

入学年度	学年	研究室名	氏名
2017	M1	椹木	梅杰 (MEI JIE).

平均 μ，分散 σ^2 の正規分布 (ガウス分布) の確率密度 $p(x)$ は次のように表せる．

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \quad (1)$$

この分布に従って独立に発生した N 個の観測データ $x_1, x_2, \cdots, x_n$ より母平均 μ，母分散 σ^2 の推定値を求めよ．そのとき，最も確からしい推定値として，以下の関数 (尤度) を最大にするような推定値，つまり最尤推定値を求めよ．

↳ likelihood

↳ maximum likelihood.

$$L(\mu, \sigma^2) = \prod_{i=1}^{N} p(x_i) \quad (2)$$

$$= \prod_{i=1}^{N} \frac{e^{-(x_i-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \quad (3)$$

$$= \frac{e^{\sum_{i=1}^{N} -(x_i-\mu)^2/2\sigma^2}}{(\sqrt{2\pi\sigma^2})^N} \quad (4)$$

$$\ln L = \left[\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}^{\frac{N}{2}}}\right)\right] - \left[\sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2}\right] \Leftarrow$$

maximum likelihood calculation:

① $\ln L(\mu, \sigma^2) = \ln \dfrac{e^{\sum_{i=1}^{N} -(x_i-\mu)^2/2\sigma^2}}{(\sqrt{2\pi\sigma^2})^N} = \left[-\frac{1}{2} N \ln(2\pi)\right] - \left[N \ln(\sigma)\right] - \left[\dfrac{\sum_{i=1}^{N}(x_i-\mu)^2}{2\sigma^2}\right]$

② $\dfrac{\partial \ln L}{\partial \mu} = \dfrac{\sum_{i=1}^{N}(x_i-\mu)}{2\sigma^2} = 0 \Rightarrow \hat{\mu} = \dfrac{\sum_{i=1}^{N} x_i}{N}$

③ $\dfrac{\partial \ln L}{\partial \sigma} = -\dfrac{N}{\sigma} + \dfrac{\sum_{i=1}^{N}(x_i-\mu)^2}{\sigma^2} = 0 \Rightarrow \hat{\sigma} = \sqrt{\dfrac{\sum_{i=1}^{N}(x_i-\hat{\mu})^2}{N}}$

In short, when $(\mu, \sigma) = (\hat{\mu}, \hat{\sigma})$, likelihood function reaches maximum.

Optimization Problem.

1. ① Constrained optimization problems consider the problem of optimizing an objective function subjective to constraints on the variables. In general term,

$$\min / \max \quad f(X) = f(x_1, x_2, \cdots x_n)$$
$$\text{st.} \quad g_1(X) = g_1(x_1, x_2, \cdots x_n) = b_1$$
$$\vdots$$
$$g_m(X) = g_m(x_1, x_2, \cdots x_n) = b_m$$

One standard way of attempting to solve the problem is via the method of Lagrange Multipliers.

$$L(X, \lambda) = f(X) + \lambda_1(b_1 - g_1(X)) + \cdots + \lambda_m(b_m - g_m(X))$$

Now optimize $L(X, \lambda)$ as a unconstrained optimization problem in the original independent variables X and a new variable λ.

$$\begin{cases} \dfrac{\partial L}{\partial x_i} = 0 \Rightarrow f_i - \sum_j \lambda_j g_i^j = 0 & (i = 1, 2, \cdots n) \\ \dfrac{\partial L}{\partial \lambda_j} = 0 \Rightarrow b_j - g^j(X) = 0 & (j = 1, 2, \cdots m) \end{cases}$$

where $g_i^j = \dfrac{\partial g^j(X)}{\partial x_i}$. And then obtain the stationary point X_i^* and λ^*.

② Except the equality constraints problem, the Lagrange Multipliers Method can be applied to the inequality one as well.

$$\max \; f(X) \qquad\qquad \min \; f(X)$$
$$\text{st.} \; g_j(X) \le b_j \qquad\qquad \text{st.} \; g_j(X) \geq b_j.$$
$$\Rightarrow g_j(X) + z^2 = b_j \qquad\qquad \Rightarrow g_j(X) - z^2 = b_j.$$

After transfering the inequality constaints into equality, we can optimize $L(X, \lambda, z)$ with the same method once again. Noted that there are some Kuhn-Tucker necessary conditions needing to be satisfied.

$$\begin{cases} f_i - \sum_j \lambda_j g_i^j = 0 \\ \lambda_j g_j(X) = 0 \\ \lambda_j \geq 0 \\ g_j(X) \le b_j \end{cases} \qquad\qquad \begin{cases} f_i - \sum_j \lambda_j g_i^j = 0 \\ \lambda_j g_j(X) = 0 \\ \lambda_j \geq 0 \\ g_j(X) \le b_j. \end{cases}$$

Besides, with the second order conditions / the bordered Hessian, we can further know whether stationary points are maximum or minimum.

2. ① Considering it as a linear program in standard form that:

$$\min \quad C^T X$$
$$\text{st.} \quad AX = b \quad \text{and} \quad X \geq 0.$$

where $\mathbb{X} = [X_1, X_2 \cdots X_n]^T$ $\mathbb{C} = [C_1, C_2 \cdots C_n]^T$

$$\mathbb{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \Rightarrow \text{Rank}(\mathbb{A}) = m \quad (m<n)$$

$\mathbb{b} = [b_1, b_2 \cdots b_m]^T \geq 0$.

And the basic solution for standard form is shown as follow:

① let $\mathbb{B}$ be a square matrix whose columns are m linearly independent columns of $\mathbb{A}$

② let $\mathbb{D}$ be the $m \times (n-m)$ matrix whose colums ar the remaining columns of $\mathbb{A}$

③ $\mathbb{A} = [\mathbb{B} \; \mathbb{D}]$ and cause matrix $\mathbb{B}$ in nonsingular, thus we can solve the equation $\mathbb{B} X_B = \mathbb{b}$ for the m-vector X_B.

④ let $\mathbb{X}$ be the n-vector whose first m components are equal to X_B and the remaining components are equal to zero, that is, $\mathbb{X} = [X_B^T, 0^T]^T$ as the solution to $\mathbb{A}\mathbb{X} = \mathbb{b}$.

For example, $\mathbb{A} = \begin{bmatrix} 1 & 1 & -1 & 4 \\ 1 & -2 & -1 & 1 \end{bmatrix}$ $\mathbb{b} = \begin{bmatrix} 8 \\ 2 \end{bmatrix}$, then

$\mathbb{X} = [6, 2, 0, 0]^T$ is the basic solution with respect to $\mathbb{B} = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix}$

$\mathbb{X} = [0, 0, 0, 2]^T$ is the basic solution with respect to $\mathbb{B} = \begin{bmatrix} -1 & 4 \\ -1 & 1 \end{bmatrix}$

...

Noted that a vector $\mathbb{X}$ satisfying $\mathbb{A}\mathbb{X} = \mathbb{b}$ and $\mathbb{X} \geq 0$ is said to be a feasible solution.

② Considering it as a nonlinear least-squares problem that:

$$\min \|\mathbb{A}\mathbb{X} - \mathbb{b}\|^2 \Rightarrow \|\mathbb{A}\mathbb{X} - \mathbb{b}\|^2 \geq \|\mathbb{A}\mathbb{X}^* - \mathbb{b}\|^2$$

where vector $\mathbb{X}^*$ is the least-squares solution to $\mathbb{A}\mathbb{X} = \mathbb{b}$.

Then, since $\mathbb{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$, $\text{Rank}(\mathbb{A}) = n$ if and only if $\text{Rank}(\mathbb{A}^T\mathbb{A}) = n$, which means the square matrix $\mathbb{A}^T\mathbb{A}$ is nonsingular and inversable.

$$\mathbb{X}^* = (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\mathbb{b} \quad \text{from} \quad \mathbb{A}^T\mathbb{A}\mathbb{X} = \mathbb{A}^T\mathbb{b}.$$

3. Considering it as a constrained optimization problem that:

$$\min_{\mathbb{U}(k)} J = \sum_{k=0}^{N} L(\mathbb{X}(k), \mathbb{U}(k))$$

$$\text{s.t.} \quad \mathbb{X}(k+1) = f(\mathbb{X}(k), \mathbb{U}(k))$$

$$\mathbb{X}(0) = \mathbb{X}_0.$$

Then transfer the object function into a general form as follow:

$$J_N[X(M)] = \Phi[X(N)] + \sum_{k=M}^{N-1} L[X(k), u(k)].$$

Noted that M is the initial point and N is the final point.

$$\Rightarrow J_N[X(0)] = L[X(0), u(0)] + \cdots + L[X(N-1), u(N-1)] + \Phi[X(N)]$$

$$= L[X(0), u(0)] + \cdots + L[X(0), u(0), \cdots u(N-1)] + \Phi[X(0), u(0) \cdots u(N-1)]$$

$$= J_N[X(0), u(0) \cdots u(N-1)]$$

$$\Rightarrow J_N^*[X(0)] = \min_{u(0)\cdots u(N-1)} J_N[X(0), u(0) \cdots u(N-1)]$$

assuming that $u^*(0)$ is given, then $X(1) = f[X(0), u^*(0)]$

$$\Rightarrow J_{N-1}^*[X(1)] = \min_{u(1)\cdots u(N-1)} J_{N-1}[X(1), u(1) \cdots u(N-1)]$$

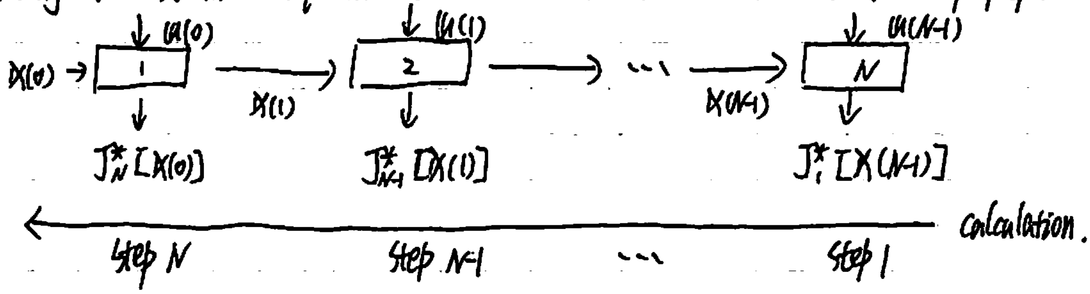
$$\Rightarrow J_N^*[X(0)] = \min_{u(0)} \{L[X(0), u(0)] + J_{N-1}^*[X(1)]\}$$

more generally, we can gain that

$$J_{N-k}^*[X(k)] = \min_{u(k)} \{L[X(k), u(k)] + J_{N-(k+1)}^*[X(k+1)]\}$$

$$X(k+1) = f[X(k), u^*(k)] \quad (k \in [0, N-1])$$

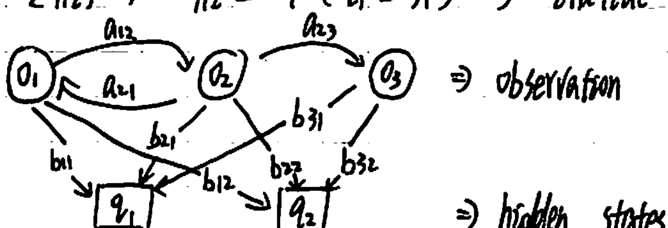
According to Bellman Equation, it can be solved via backward propagation.



4. Viterbi Algorithm is the method to find most probable sequence of hidden states within Hidden Markov Models via Dynamic Programming.

① we define a HMM as $\langle A, B, \pi \rangle$, where

$$\begin{cases} A = \{a_{ij}\}, & a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i) \Rightarrow \text{State transition probability} \\ B = \{b_{ik}\}, & b_{ik} = P(O_t = V_k \mid q_t = S_i) \Rightarrow \text{Observation symbol probability.} \\ \pi = \{\pi_i\}, & \pi_i = P(q_1 = S_i) \Rightarrow \text{initial State probability.} \end{cases}$$



② Then define the decoding problem as: Given the observation sequence

$O = [O_1, O_2, \cdots, O_T]$ and a model $\langle A, B, \pi \rangle$, to choose a corresponding state sequence $Q = [q_1, q_2, \cdots q_T]$ which is optimal in some sence.

③ the Viterbi Algorithm.

when $t=1$, $\delta_1(i) = P(q_1 = s_i) \cdot P(O_1 = v_1 \mid q_1 = s_i) = \pi_i \cdot b_{i1}$

when $t>1$, $\delta_{t+1}(i) = \max_{q_1 \cdots q_{t+1}} P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t) = [\max_j \delta_t(j) a_{ji}] \cdot b_{it}$

set $\phi_{t+1}(i) = \operatorname{argmax}_j [\delta_t(j) a_{ji}]$, then can gain the optimum with dynamic programming.

④ Since result of t+1 is based on the one of t, there will be a accumulated bias. But with the help of $\phi_{t+1}(i)$ and back propogation, Vitebi algorithm can remove this bias.

5. Value Iteration with Dynamic Programming.

⇒ given an MDP $\langle S, A, P, R, \gamma, H \rangle$ to find the optimal policy π^*

⇒ $V^*(s) = \max_\pi \mathbb{E}[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1}) \mid \pi, s_0 = s]$

⇒ start with $V_0^*(s) = 0$ for all s

for $k = 1, \cdots, H$:

for all states in S:

$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma V_{k-1}^*(s')]$

$\pi_k^*(s) \leftarrow \operatorname{argmax}_a \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma V_{k-1}^*(s')]$

⇒

			+1
	▨		−1

$s \to s'$ Probability = 0.8

reward discount = 0.9

⇒

0	0	0	+1
0	▨	0	−1
0	0	0	0

step 1

⇒

0	0	0.72	+1
0	▨	0	−1
0	0	0	0

$0.72 = 0.8 \times [0 + 0.9 \times 1]$

step 2

⇒

0	0.52	0.78	+1
0	▨	0.43	−1
0	0	0	0

$0.52 \approx 0.8 \times [0 + 0.9 \times 0.72]$

$0.43 \approx 0.8 \times [0 + 0.9 \times 0.72] + 0.1 \times [0 + 0.9 \times -1]$

step 3

⋯ ⋯ ⇒

0.64	0.74	0.85	+1
0.57	▨	0.57	−1
0.49	0.43	0.46	0.28

step H=100.

6. Considering it as a linear regression problem, that:

$$\begin{cases} \text{regression function} \Rightarrow \hat{y}_i = a x_i + b \\ \text{cost function} \Rightarrow J = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} [y_i - (ax_i + b)]^2 \end{cases}$$

then
$$\begin{aligned} J &= \sum_{i=1}^{n} [y_i - (ax_i + b)]^2 \\ &= [y_1 - (ax_1 + b)]^2 + [y_2 - (ax_2 + b)]^2 + \cdots + [y_n - (ax_n + b)]^2 \\ &= [y_1^2 - 2(ax_1 + b)y_1 + (ax_1 + b)^2] + \cdots + [y_n^2 - 2(ax_n + b)y_n + (ax_n + b)^2] \\ &= [y_1^2 + \cdots + y_n^2] - 2a(x_1 y_1 + \cdots x_n y_n) - 2b(y_1 + \cdots + y_n) + a^2(x_1^2 + \cdots + x_n^2) \\ &\quad + 2ab(x_1 + \cdots + x_n) + nb^2 \end{aligned}$$

set $\overline{y^2} = (y_1^2 + \cdots + y_n^2)/n$, $\overline{xy} = (x_1 y_1 + \cdots + x_n y_n)/n$, then

$$J = n\overline{y^2} - 2an\overline{xy} - 2bn\bar{y} + a^2 n \overline{x^2} + 2abn\bar{x} + nb^2$$

when $\frac{\partial J}{\partial a} = \frac{\partial J}{\partial b} = 0$, J reaches optimum and that a and b are what we ~~wa~~

$$\begin{cases} \frac{\partial J}{\partial a} = -2n\overline{xy} + 2an\overline{x^2} + 2bn\bar{x} = 0 \\ \frac{\partial J}{\partial b} = -2n\bar{y} + 2an\bar{x} + 2nb = 0 \end{cases} \Rightarrow \begin{cases} a = \frac{\bar{x}\bar{y} - \overline{xy}}{(\bar{x})^2 - \overline{x^2}} \\ b = \bar{y} - a\bar{x} \end{cases}$$

so $(-\frac{C}{B}) = b$ and $(-\frac{C}{A}) = a$

7. Considering it as an extended Kalman filter, that

$$\begin{cases} X(k+1) = f[X(k), U(k), W(k)] & \Rightarrow \text{state transition function} \\ Y(k) = h[X(k)] + V(k) & \Rightarrow \text{measurement function} \end{cases}$$

where $W(k)$ and $V(k)$ are the unknown noise. then

$$\begin{cases} \hat{X}(k+1) = f[\hat{X}(k), U(k), 0] \\ \hat{Y}(k) = h[\hat{X}(k)] + 0 \end{cases} \Rightarrow \text{approximation function.}$$

$$\Rightarrow \begin{cases} X(k+1) = \hat{X}(k+1) + A[X(k) - \hat{X}(k)] + B\,W(k) \\ Y(k) = \hat{Y}(k) + C[X(k) - \hat{X}(k)] + D\,V(k) \end{cases}$$

$$\Rightarrow A_{ij} = \frac{\partial f_i}{\partial x_j}, \quad B_{ij} = \frac{\partial f_i}{\partial w_j}, \quad C_{ij} = \frac{\partial h_i}{\partial x_j}, \quad D_{ij} = \frac{\partial h_i}{\partial v_j}$$

$$\Rightarrow e_{xk} = X(k) - \hat{X}(k) \qquad e_{yk} = Y(k) - \hat{Y}(k) \qquad \Rightarrow e_k = K_k \cdot e_{yk}$$

$$\begin{aligned} &= A[X(k) - \hat{X}(k)] + B W(k) = C[X(k) - \hat{X}(k)] + D V(k) \\ &\approx A[X(k) - \hat{X}(k)] + \varepsilon(k) \approx C[X(k) - \hat{X}(k)] + \eta(k) \end{aligned}$$

Noted that $\varepsilon(k) \sim N(0, BQB)$ $\quad \eta(k) \sim N(0, DRD)$

$$\Rightarrow X(k) \leftarrow \hat{X}(k) + e_k = \hat{X}(k) + K_k \cdot e_{yk} = \hat{X}(k) + K_k \cdot [Y(k) - \hat{Y}(k)]$$

set $P_k = E[(X(k) - \hat{X}(k)) \cdot (X(k) - \hat{X}(k))^T]$, then

$$① \begin{cases} \hat{X}^-(k) \leftarrow f[\hat{X}(k-1), U(k-1), 0] \\ P_k^- \leftarrow A_k P_{k-1} A_k^T + B_k Q_{k-1} B_k^T \end{cases}$$

$$② \begin{cases} K_k \leftarrow P_k^- C_k^T (C_k P_k^- C_k^T + D_k R_k D_k^T)^{-1} \\ \hat{X}_k \leftarrow \hat{X}_k^- + K_k \{ Y_k - h[\hat{X}^-(k)] \} \\ P_k \leftarrow (I - K_k C_k) \cdot P_k^- \end{cases}$$

8, ① Considering a example as Random Constant Estimation.

$$\begin{cases} x_k = A x_{k-1} + B u_{k-1} + w_k = x_{k-1} + w_k \\ y_k = x_k + v_k \end{cases}$$

so
$$\begin{cases} \hat{x}_k^- = \hat{x}_{k-1} \\ P_k^- = P_{k-1} + Q \end{cases}$$

$$\begin{cases} K_k = P_k^- (P_k^- + R)^{-1} \\ \hat{x}_k = \hat{x}_k^- + K_k (y_k - \hat{x}_k^-) \\ P_k = (1 - K_k) P_k^- \end{cases}$$

② As assignment 7 shown, it can be applied to kalman Filter as the Extended kalman Filter.

$\hat{x}_k^-$ is the prediction.
P_k^- is the Priori estimate error covariance
K_k is the gain to minimize P_k^-
$\hat{x}_k$ is the post prediction
P_k is the postpriori estimate error covariance

9. As the assignment 5 shown, given a grid world with reward, we can gain a optimal policy via Dynamic Programming. In other words, within Reinforcement learning, Dynamic Programming can be used to calculate the value of a state or the action-value of a state after being given a Markov Model. However, In realistic, it is impossible to pre-own a model or it will cost a lot of computation to handle a pratical problem. So model-free Reinforcement learning is proposed with the sampling method.

As for the prediction and error estimation, approximation function will help. With a large state space, it will cost a lot of memory to store all the states and their values. So an idea of using approximation function to represent the states and their values is put forward.

$\Rightarrow$ Q-table: $Q_{k+1}(s,a) \leftarrow (1-\alpha) Q_k(s,a) + \alpha [R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$

Approximate Q-learning: $\theta_{k+1} \leftarrow \theta_k - \alpha \cdot \nabla_\theta \mathbb{E}_{s' \sim P(s'|s,a)} [(Q_\theta(s,a) - (R(s,a,s') + \gamma \max_{a'} Q_k(s',a'))]^2$

$\Rightarrow$ updat the parameter θ instead of action value itself with the help of least square / gradient decent.

At the same time, the Markov Model can be approximated as well, which is called Dyna Method in Reinforcement learning.

Modification

1. The Numerical Solution of Constrained Optimization problems:

① Linear ⇒ { Simplex Method; Duality; Non-Simplex Method ⇒ { Khachiyan's Method; Affine Scaling Method; Karmarkar's Method } }

② Nonlinear ⇒ { equality constraints ⇒ Largrange condition. ; Inequality constraints ⇒ Karush-Kuhn-Tucker Condition. }

⇒ Unconstrained optimization problem:

① one dimentional search method
② gradient method
③ newton's method
④ conjugate direction method
⑤ quasi-newton method.

6. The Maximum Likelihood Method for Linear Regression:

$$\begin{cases} \hat{Y} = W^T \cdot X \\ Y = W^T \cdot X + \Sigma \end{cases} \qquad J(\theta) = \frac{1}{2} \sum_{i=1}^{n} [(\hat{Y}_i - y_i')^2]$$

$$\Rightarrow P(y_i | x_i, w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{y_i - \hat{y}_i}{2\sigma^2}\right]$$

$$\Rightarrow L' = \prod_{i=1}^{N} P(y_i | x_i, w) \Rightarrow L = \log L' = \frac{N}{2} \log 2N - N \log \sigma - \frac{1}{2\sigma^2}(y_i - \hat{y}_i)^2$$

$$\Rightarrow \frac{\partial L}{\partial W} = \frac{1}{\sigma^2} \sum_{i=1}^{N} x_i (y_i - x_i W) = 0 \Rightarrow \frac{\partial L}{\partial W} = \frac{1}{\sigma^2} (X^T Y - X^T X W) = 0$$

$$\Rightarrow W = (X^T X)^{-1} X^T Y. \Rightarrow W = \left[-\frac{A}{B} \quad -\frac{C}{B}\right]^T.$$