Cours MAP434 : Contrôle de modèles dynamiques

Séance 7, 30 mai 2018

Programmation dynamique en temps discret

Exercice I. Système LQ en temps discret

On considère le système linéaire en temps discret

$$x_{m+1} = \mathcal{A}x_m + \mathcal{B}u_m \quad \text{pour } 0 \le n \le m \le N-1, \quad x_n = x \in \mathbb{R}^d,$$

où n, m, N sont des entiers, $\mathcal{A}$ est une matrice $d \times d$, $\mathcal{B}$ est une matrice $d \times k$, et $u_m \in \mathbb{R}^k$. On considère la fonction valeur

$$V_n(x) = \min_{u \in (\mathbb{R}^k)^{N-n}} \left\{ \sum_{m=n}^{N-1} \left(\frac{1}{2}u_m^\dagger R u_m + \frac{1}{2}x_m^\dagger Q x_m \right) + \frac{1}{2}x_N^\dagger D x_N \right\},$$

où R est une matrice symétrique $k \times k$ définie positive, Q et D sont des matrices symétriques $d \times d$ semi-définies positives. On pose $V_N(x) = \frac{1}{2}x^\dagger D x$.

Question 1. Écrire l'équation de programmation dynamique satisfaite par V_n.

Correction: On a

$$\begin{aligned} V_n(x) &= \min_{u \in \mathbb{R}^k} \left\{ \frac{1}{2}u^\dagger R u + \frac{1}{2}x^\dagger Q x \right. \\ &\quad \left. + \min_{u \in (\mathbb{R}^k)^{N-n-1}} \sum_{m=n+1}^{N-1} \left(\frac{1}{2}u_m^\dagger R u_m + \frac{1}{2}x_m^\dagger Q x_m \right) + \frac{1}{2}x_N^\dagger D x_N \right\} \\ &= \min_{u \in \mathbb{R}^k} \left\{ \frac{1}{2}u^\dagger R u + \frac{1}{2}x^\dagger Q x + V_{n+1}(\mathcal{A}x + \mathcal{B}u) \right\}. \end{aligned}$$

Question 2. Montrer par récurrence (rétrograde) que $V_n(x) = \frac{1}{2}x^\dagger P_n x$ où P_n est une matrice symétrique semi-définie positive définie $d \times d$ régie par l'équation de Riccati discrète

$$\begin{aligned} &P_N = D, \\ &P_n = Q + \mathcal{A}^\dagger P_{n+1}\mathcal{A} - (\mathcal{A}^\dagger P_{n+1}\mathcal{B})(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}(\mathcal{A}^\dagger P_{n+1}\mathcal{B})^\dagger, \quad \forall 0 \le n \le N-1. \end{aligned}$$

Calculer le contrôle optimal.

Correction: Pour $n = N$, le résultat est évident. Supposons qu'il est vrai pour $n+1$ avec $n < N$. Alors, par l'équation de programmation dynamique obtenue à la question précédente, il vient

$$\begin{aligned} V_n(x) &= \min_{u \in \mathbb{R}^k} \left\{ \frac{1}{2}u^\dagger R u + \frac{1}{2}x^\dagger Q x + \frac{1}{2}(\mathcal{A}x + \mathcal{B}u)^\dagger P_{n+1}(\mathcal{A}x + \mathcal{B}u) \right\} \\ &= \frac{1}{2}x^\dagger (Q + \mathcal{A}^\dagger P_{n+1}\mathcal{A})x + \min_{u \in \mathbb{R}^k} \left\{ \frac{1}{2}u^\dagger (R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})u + x^\dagger (\mathcal{A}^\dagger P_{n+1}\mathcal{B})u \right\}. \end{aligned}$$

La valeur optimale du contrôle (sous forme de feedback) est donnée par

$$u_n^*(x) = -(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger P_{n+1}\mathcal{A}x,$$

et en reportant dans l'expression de la fonction valeur, on obtient

$$\begin{aligned} V_n(x) &= \frac{1}{2}x^\dagger(Q + \mathcal{A}^\dagger P_{n+1}\mathcal{A})x - \frac{1}{2}x^\dagger(\mathcal{B}^\dagger P_{n+1}\mathcal{A})^\dagger(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}(\mathcal{B}^\dagger P_{n+1}\mathcal{A})x \\ &= \frac{1}{2}x^\dagger P_n x, \end{aligned}$$

avec la matrice P_n satisfaisant les relations données dans l'énoncé. Enfin, on vérifie facilement que la matrice P_n est symétrique semi-définie positive.

Question 3. On suppose dans cette question que D est définie positive, et que $\ker(A)\cap\ker(Q) = \{0\}$. Montrer que P_n est également donnée par l'équation suivante :

$$P_N = D, \qquad P_n = Q + \mathcal{A}^\dagger\left(\mathcal{B}R^{-1}\mathcal{B}^\dagger + P_{n+1}^{-1}\right)^{-1}\mathcal{A}, \quad \forall 0 \le n \le N-1.$$

Correction: On a $P_N = D$ qui est bien symétrique définie positive. De plus, on a

$$P_n = Q + \mathcal{A}^\dagger[P_{n+1} - P_{n+1}\mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger P_{n+1}]\mathcal{A}.$$

En supposant que P_{n+1} est inversible, on obtient

$$\begin{aligned} &(P_{n+1} - P_{n+1}\mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger P_{n+1})(\mathcal{B}R^{-1}\mathcal{B}^\dagger + P_{n+1}^{-1}) \\ &= I + P_{n+1}[\mathcal{B}R^{-1}\mathcal{B}^\dagger - \mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger P_{n+1}\mathcal{B}R^{-1}\mathcal{B}^\dagger - \mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger] \\ &= I + P_{n+1}[\mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger - \mathcal{B}(R + \mathcal{B}^\dagger P_{n+1}\mathcal{B})^{-1}\mathcal{B}^\dagger] = I. \end{aligned}$$

Ceci démontre donc la formule annoncée. De plus, en utilisant cette formule, on voit que $x^\dagger P_n x \ge 0$, avec égalité si et seulement si $x^\dagger Q x = 0$ et $x^\dagger\mathcal{A}^\dagger[\mathcal{B}R^{-1}\mathcal{B}^\dagger + P_{n+1}^{-1}]^{-1}\mathcal{A}x = 0$. La première égalité implique $Qx = 0$ et la troisième implique $\mathcal{A}x = 0$, ce qui, par hypothèse, implique $x = 0$. Donc P_n est bien définie positive.

Question 4. Pour la suite de l'exercice, on considère le cas particulier où $d = k = 1$, $\mathcal{A} = 1$, $\mathcal{B} = \Delta t$ avec $N\Delta t = 1$, $R = Q = \Delta t$ et $D = 1$. La matrice P_n est à valeurs scalaires et on la notera μ_n. Écrire l'équation de la dynamique discrète donnant x_{n+1} en fonction de x_n, le controle optimal $u_n^*(x)$ comme un feedback et la relation de récurrence rétrograde pour le coefficient μ_n (on utilisera la question précéente).

Correction: On obtient $x_{n+1} = x_n + \Delta t u_n$,

$$u_n^*(x) = -\frac{\mu_{n+1}}{1 + \mu_{n+1}\Delta t}x,$$

et

$$\mu_N = 1, \qquad \mu_n = \Delta t + \frac{\mu_{n+1}}{1 + \mu_{n+1}\Delta t}.$$

On notera que $\mu_n > 0$ pour tout n.

Question 5. On considère le cas continu avec le système de contrôle linéaire $\dot{x}_u(t) = u(t)$, pour tout $t \in [0,T]$ avec $T = 1$, $x_u(0) = x_0$, et la fonctionnelle à minimiser est

$$J(u) = \frac{1}{2}\int_0^1 (x_u(t)^2 + u(t)^2)\,\mathrm{d}t + \frac{1}{2}x_u(1)^2.$$

En utilisant le PMP, trouver la trajectoire optimale et le contrôle optimal comme un feedback.

Correction: Par convexité (cf. la proposition 5.12), le PMP fournit une condition nécessaire et suffisante d'optimalité. En introduisant l'état adjoint $p : [0,T] \to \mathbb{R}$, on obtient le système

$$\begin{aligned} \frac{d\overline{x}}{dt}(t) &= \overline{u}(t), \qquad && \overline{x}(0) = x_0, \\ \frac{d\overline{p}}{dt}(t) &= -\overline{x}(t), && \overline{p}(1) = \overline{x}(1), \\ \overline{u}(t) &= -\overline{p}(t). \end{aligned}$$

Ce qui donne $\overline{p}(t) = \overline{x}(t) = x_0 e^{-t}$, la commande optimale sous forme de feedback $\tilde{u}(x) = -x$ (si bien que $\overline{u}(t) = -x_0 e^{-t}$), et l'équation de Riccati $\dot{\mu}(t) = \mu(t)^2 - 1$, $\mu(1) = 1$ dont la solution est $\mu(t) = 1$ pour tout $0 \le t \le 1$ (on retrouve que $\overline{p}(t) = \overline{x}(t)$).

Question 6. On revient au cas discret et on fait tendre Δt vers 0 (ou de manière équivalente N vers l'infini). Montrer que $1 \le \mu_n \le 1+\Delta t$, que le contrôle optimal est tel que $u_n^*(x) = -x + O(\Delta t)$ et que la trajectoire optimale est telle que $x_n = x_0 e^{-n\Delta t} + O(\Delta t)$.

Correction: Montrons par récurrence rétrograde que pour tout $0 \le n \le N$, $1 \le \mu_n \le 1 + (N - n)\Delta t^2$ (d'où $\mu_n \le 1 + \Delta t$). On a bien sûr $\mu_N = 1$, et si l'hypothèse est vraie pour $n + 1 \le N$, en observant que

$$\mu_n = \Delta t + \frac{\mu_{n+1}}{1 + \mu_{n+1}\Delta t} = \frac{\mu_{n+1} + \Delta t + \mu_{n+1}\Delta t^2}{1 + \mu_{n+1}\Delta t},$$

il vient

$$\mu_n \ge \frac{1 + \Delta t + \Delta t^2}{1 + (1 + (N - n - 1)\Delta t^2)\Delta t} = \frac{1 + \Delta t + \Delta t^2}{1 + \Delta t + (N - n - 1)\Delta t^3}.$$

Comme $(N - n - 1) \le N$, on a donc $(N - n - 1)\Delta t^3 \le N\Delta t^3 = \Delta t^2$. On en déduit que $\mu_n \ge 1$. Par ailleurs, on constate que

$$\begin{aligned} \mu_n &\le \frac{1 + (N - n - 1)\Delta t^2 + \Delta t + (1 + (N - n - 1)\Delta t^2)\Delta t^2}{1 + \Delta t} \\ &= 1 + \Delta t^2 \frac{N - n + (N - n - 1)\Delta t^2}{1 + \Delta t} \\ &= 1 + (N - n)\Delta t^2 \frac{1 + (1 - 1/(N - n))\Delta t^2}{1 + \Delta t} \\ &\le 1 + (N - n)\Delta t^2 \le 1 + \Delta t. \end{aligned}$$

En conclusion, on a prouvé que $1 \le \mu_n \le 1 + \Delta t$ pour tout n. Comme on avait montré que le contrôle optimal est tel que $u_n^*(x) = -\frac{\mu_{n+1}}{1+\mu_{n+1}\Delta t}x$, on en déduit que $u_n^*(x) = -x + O(\Delta t)$. Enfin, concernant la trajecoire optimale, comme

$$x_{n+1} = x_n + \Delta t u_n = x_n - \Delta t \frac{\mu_{n+1}}{1 + \mu_{n+1}\Delta t} x_n = \frac{x_n}{1 + \mu_{n+1}\Delta t},$$

on déduit des estimations sur μ_{n+1} que

$$x_n = x_{n+1}(1 + \mu_{n+1}\Delta t) = x_{n+1}(1 + \Delta t + O(\Delta t^2)).$$

Par suite,

$$x_0 = x_N \left(1 + \frac{1}{N} + O\left(\frac{1}{N^2}\right)\right)^N = x_N e^{N \ln(1 + \frac{1}{N} + o(N^{-1}))} = e^1 x_N + O(\Delta t).$$

Par suite, pour tout $0 \leq n \leq N$, il vient

$$x_n = x_0 \left(1 + \frac{1}{N} + o(N^{-1})\right)^{-n} = x_0 e^{-n \ln(1 + \frac{1}{N} + o(N^{-1}))} = x_0 e^{-n\Delta t} + O(\Delta t).$$

Question 7. En supposant que μ_n est une approximation de $\mu(n\Delta t)$, où $\mu : [0,T] \to \mathbb{R}$ est une fonction régulière, montrer que la fonction μ satisfait, à $O(\Delta t)$ près, l'équation de Riccati continue aux instants discrets $t_n = n\Delta t$.

Correction: On effectue un développement limité en écrivant $\mu_{n+1} = \mu(t_n) + \Delta t \mu'(t_n) + o(\Delta t)$ et on injecte ce développement dans l'équation de Riccati discrète satisfaite par la suite des μ_n ; il vient

$$\begin{aligned}
\mu_n &= \Delta t + \frac{\mu_{n+1}}{1 + \mu_{n+1}\Delta t} \\
&= \Delta t + \frac{\mu(t_n) + \mu'(t_n)\Delta t + o(\Delta t)}{1 + \Delta t(\mu(t_n) + \mu'(t_n)\Delta t + o(\Delta t))} \\
&= \Delta t + (\mu(t_n) + \mu'(t_n)\Delta t + o(\Delta t))(1 - \Delta t \mu(t_n) + o(\Delta t)) \\
&= \mu(t_n) + \Delta t(1 + \mu'(t_n) - \mu(t_n)^2) + o(\Delta t).
\end{aligned}$$

On obtient bien une approximation de l'équation de Riccati continue

$$1 + \mu'(t_n) - \mu(t_n)^2 = O(\Delta t).$$

Exercice II. Optimisation combinatoire appliquée à la génétique

On s'intéresse à comparer des séquences d'ADN : l'alphabet est composé des 4 lettres $\Sigma = \{A, T, G, C\}$, un mot w est défini par sa longueur k et une suite de k lettres prises dans l'alphabet, soit $w = a_1 a_2 \cdots a_k \in \Sigma^k$. Soit maintenant un deuxième mot $z = b_1 b_2 \cdots b_l \in \Sigma^l$, un alignement de w sur z consiste en la donnée de deux sous-suites strictement croissantes $i_1, \cdots, i_m \in \{1, \cdots, k\}$ et $j_1, \cdots, j_m \in \{1, \cdots, l\}$ avec $m \leq \min(k, l)$ (cela définit les m lettres que l'on va mettre face à face). Un alignement est noté $(i_1 \cdots i_m | j_1 \cdots j_m)$ et le score de cet alignement est défini par

$$S(i_1 \cdots i_m | j_1 \cdots j_m) = \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) - \delta(k + l - 2m).$$

Ce score est composé de la somme des gains $M(a_{i_r}, b_{j_r})$ des lettres mises face à face moins le nombre des lettres que l'on n'a pas alignées multiplié par un coefficient de pénalité δ. Dans la suite, nous prendrons $\delta = 0.5$ et $M(a,a) = 1$ pour tout $a \in \Sigma$, $M(a,b) = 0$ si $a \neq b$. Enfin, dans le cas $m = 0$, le score est pris égal à $-\delta(k+l)$. On peut représenter l'alignement en positionnant le mot w sur l'axe des x et le mot z sur l'axe des y, une flèche horizontale traduisant que l'on saute la lettre du mot de l'axe des x, une flèche verticale que l'on saute celle du mot de l'axe des y et une flèche diagonale que l'on aligne la lettre suivante des deux mots.

Question 1. Soit $w = ATGA$ et $z = AGATT$. Calculer $S(1,3,4|1,2,4)$ et représenter cet alignement. Que peut-on dire de l'alignement $(1,3,4|1,3,4)$?

Correction: L'alignement s'écrit :

$$\begin{array}{cccccc}
A & T & G & & A & \\
| & & | & & | & \\
A & & G & A & T & T
\end{array}$$

et le score correspondant $S(1,3,4|1,2,4) = M(A,A) + M(G,G) + M(A,T) - 0.5 * 3 = 0.5$. Il se représente comme suit :

$$\begin{array}{cccccccccc}
T & & & & & & & & & 0.5 \\
 & & & & & & & & & \uparrow \\
T & & & & & & & & & 1. \\
 & & & & & & & & \nearrow & \\
A & & & & & & & 1. & & \\
 & & & & & & & \uparrow & & \\
G & & & & & & & 1.5 & & \\
 & & & & & & \nearrow & & & \\
A & & & 1. & \rightarrow & 0.5 & & & & \\
 & & \nearrow & & & & & & & \\
 & 0. & & & & & & & & \\
 & & & A & & T & & G & & A
\end{array}$$

Dans le cas de l'alignement $(1,3,4|1,3,4)$, on ne distinguera pas les alignements

$$\begin{array}{cccccc}
A & T & & G & A & \\
| & & & | & | & \\
A & & G & A & T & T
\end{array}
\quad \text{et} \quad
\begin{array}{cccccc}
A & & T & G & A & \\
| & & & | & | & \\
A & G & & A & T & T
\end{array}$$

soit dans le plan les deux tracés

$$\begin{array}{cccccccccc}
T & & & & & & & & & -0.5 \\
 & & & & & & & & & \uparrow \\
T & & & & & & & & & 0. \\
 & & & & & & & & \nearrow & \\
A & & & & & & & 0. & & \\
 & & & & & & \nearrow & & & \\
G & & & 0.5 & \rightarrow & 0. & & & & \\
 & & & \uparrow & & \uparrow & & & & \\
A & & & 1. & \rightarrow & 0.5 & & & & \\
 & & \nearrow & & & & & & & \\
 & 0. & & & & & & & & \\
 & & & A & & T & & G & & A
\end{array}$$

Question 2. On définit le score des deux mots $(w = a_1 \cdots a_k, z = b_1 \cdots b_l)$ par

$$S(w,z) = \max_m \max_{\substack{i_1 \cdots i_m \\ j_1 \cdots j_m}} S(i_1 \cdots i_m | j_1 \cdots j_m),$$

où le maximum est pris sur l'ensemble des alignements et l'on cherche le ou les alignements optimaux. Pour tout $1 \leq K \leq k$ et $1 \leq L \leq l$, on définit

$$S_{K,L} = S(a_1 \cdots a_K, b_1 \cdots b_L)$$

avec $w_K = a_1 \cdots a_K$, respectivement $z_L = b_1 \cdots b_L$, le mot w tronqué à ses K premières lettres, respectivement le mot z à ses L premières lettres. Ecrire une équation de programmation dynamique donnant $S_{K,L}$ en fonction de $S_{K-1,L}$, $S_{K,L-1}$ et $S_{K-1,L-1}$ pour $K, L \geq 1$. En déduire un algorithme permettant de trouver l'alignement optimal en effectuant $O(kl)$ comparaisons.

Correction: Rappelons que si on n'aligne aucune des lettres entre deux mots, le score obtenu est égal à $-\delta(K+L)$ avec K et L les longueurs respectives des deux mots. On peut donc définir $S_{0,0} = 0$, $S_{K,0} = -\delta K$ et $S_{0,L} = -\delta L$. On a de plus, dès que $m > 0$,

$$S(i_1 \cdots i_m | j_1 \cdots j_m) = -\delta(K+L) + 2\delta m + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) > -\delta(K+L).$$

L'alignement optimal n'est donc jamais obtenu pour un alignement nul.

Considérons un alignement optimal pour (w_K, z_L), $m > 0$ sa longueur et $i_1 \cdots i_m$, $j_1, \cdots j_m$ les suites optimales. Nous avons plusieurs cas de figure : on considère $K \geq 2$ et $L \geq 2$ pour commencer.

Soit $i_m = K$ et $j_m = L$ (on aligne les deux dernières lettres), alors

$$\begin{aligned}
S_{K,L} = S(i_1 \cdots i_m | j_1 \cdots j_m) &= -\delta(K+L-2m) + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) \\
&= -\delta(K-1+L-1-2(m-1)) + \sum_{r=1}^{m-1} M(a_{i_r}, b_{j_r}) + M(a_K, b_L) \\
&= S(i_1 \cdots i_{m-1} | j_1 \cdots j_{m-1}) + M(a_K, b_L),
\end{aligned}$$

et d'après le principe d'optimalité de Bellman, le sous alignement $(i_1 \cdots i_{m-1} | j_1 \cdots j_{m-1})$ est optimal et son score est égal à $S_{K-1,L-1}$.

Soit $i_m = K$ et $j_m \neq L$ (la lettre L de z_L n'est pas alignée), alors

$$\begin{aligned}
S_{K,L} = S(i_1 \cdots i_m | j_1 \cdots j_m) &= -\delta(K+L-2m) + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) \\
&= -\delta(K+L-1-2m) - \delta + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) \\
&= S(i_1 \cdots i_m | j_1 \cdots j_m) - \delta,
\end{aligned}$$

et d'après le principe d'optimalité de Bellman, le sous alignement $(i_1 \cdots i_m | j_1 \cdots j_m)$ est optimal et son score est égal à $S_{K,L-1}$.

Soit $i_m \neq K$ et $j_m = L$ (la lettre K de w_K n'est pas alignée) alors on a de manière analogue

$$S_{K,L} = S_{K-1,L} - \delta.$$

Soit $i_m \neq K$ et $j_m \neq L$ (la dernière lettre de chaque mot n'appartient pas à l'alignement optimal) : ce cas là ne peut être optimal ; en effet,

$$\begin{aligned}
S(i_1 \cdots i_m | j_1 \cdots j_m) &= -\delta(K + L - 2m) + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) \\
&= -\delta(K + L - 2(m+1)) + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) - 2\delta \\
&< -\delta(K + L - 2(m+1)) + \sum_{r=1}^{m} M(a_{i_r}, b_{j_r}) + M(a_K, b_L) \\
&= S(i_1 \cdots i_m K | j_1 \cdots j_m L),
\end{aligned}$$

et donc l'alignement $(i_1 \cdots i_m K | j_1 \cdots j_m L)$ est meilleur. On en déduit que

$$S_{K,L} = \max(S_{K-1,L-1} + M(a_K, b_L), S_{K,L-1} - \delta, S_{K-1,L} - \delta).$$

On vérifie ensuite que la formule s'étend à $K = 1$ et/ou $L = 1$:

$$S_{1,1} = M(a_1, b_1) = \max(S_{0,0} + M(a_1, b_1), S_{1,0} - \delta, S_{0,1} - \delta),$$

et par récurrence sur K,

$$\begin{aligned}
S_{K,1} = &\ -\delta(K-1) + \max_{i=1,\cdots K} M(a_i, b_1) \\
= &\ \max(-\delta(K-1) + M(a_K, b_1), -\delta(K) - \delta, \\
&\ -\delta(K-2) + \max_{i=1,\cdots K-1} M(a_i, b_1) - \delta).
\end{aligned}$$

En ordonnant les indices suivant les droites $K + L = C$ avec $0 \leq C \leq k + l$, les valeurs de $S_{K,L}$ pour une droite donnée ne dépendent que des valeurs de $S_{k,l}$ pour les droites situées dessous : on peut donc déterminer l'ensembles des kl valeurs de $S_{K,L}$ en parcourant successivement ces droites et pour chaque valeur il n'y a qu'à effectuer un test de maximum sur 3 valeurs : au total, l'algorithme nécessite de l'ordre de $O(kl)$ comparaisons.

Question 3. Calculer l'alignement optimal pour $w = ATGA$ et $z = AGATT$.

Correction: On utilise la représentation cartésienne pour dérouler l'algorithme en parcourant les droites $K + L = C$ pour des valeurs croissantes de C en ne conservant que les directions qui réalisent le maximum de

$$S_{K,L} = \max(S_{K-1,L-1} + M(a_K, b_L), S_{K,L-1} - \delta, S(K-1, L) - \delta)$$

$$
\begin{array}{cccccccccc}
T & & \to & & \to & & \to & & \to & \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
T & & \to & & \to & & \to & & \to & \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
A & -1.5 & \to & & \to & & \to & & \to & \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
G & -1. & \to & 0.5 & \to & & \to & & \to & \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
A & -0.5 & \to & 1. & \to & 0.5 & \to & & \to & \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
 & 0. & \to & -0.5 & \to & -1. & \to & -1.5 & \to & \\
 & & & A & & T & & G & & A
\end{array}
$$

et de proche en proche

$$
\begin{array}{cccccccccc}
T & -2.5 & \to & -1. & \to & 0.5 & \to & 1. & \to & 1.5 \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
T & -2. & \to & -0.5 & \to & 1. & \to & 0.5 & \to & 2. \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
A & -1.5 & \to & 0. & \to & 0.5 & \to & 1 & \to & 2.5 \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
G & -1. & \to & 0.5 & \to & 1. & \to & 1.5 & \to & 1. \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
A & -0.5 & \to & 1. & \to & 0.5 & \to & 0. & \to & -0.5 \\
 & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & \uparrow \\
 & 0. & \to & -0.5 & \to & -1. & \to & -1.5 & \to & -2. \\
 & & & A & & T & & G & & A
\end{array}
$$

On obtient que le score optimal est égal à 1.5 et est obtenu par l'alignement $(1,3,4|1,2,3)$ ou encore

$$
\begin{array}{ccccc}
A & T & G & A & \\
| & & | & | & \\
A & & G & A & TT
\end{array}
$$

On peut de façon semblable effectuer un algorithme partant du point final en baissant la valeur des lignes de niveau de $K+L$ et on obtient de même le chemin optimal.

$$
\begin{array}{cccccccccc}
T & n+2 & \to & n+1.5 & \to & n+1 & \to & n+0.5 & \to & n \\
 & & & & \nearrow & & \nearrow & & \nearrow & \uparrow \\
T & n+0.5 & \to & n & & n+0.5 & \to & n & & n+0.5 \\
 & & \nearrow & & \nearrow & & \nearrow & \uparrow & \nearrow & \uparrow \\
A & n & \to & n-0.5 & & n & & n+0.5 & & n+1 \\
 & & \nearrow & \uparrow & \nearrow & \uparrow & \nearrow & & \nearrow & \uparrow \\
G & n-1.5 & & n & & n+0.5 & \to & n & & n+1.5 \\
 & \uparrow & & & & & \nearrow & \uparrow & & \uparrow \\
A & n-1 & & n-0.5 & \to & n-1 & & n+0.5 & & n+2. \\
 & & \nearrow & & \nearrow & \uparrow & & \uparrow & \nearrow & \uparrow \\
 & n-1.5 & & n-1 & & n-0.5 & & n+1 & & n+2.5 \\
 & & & A & & T & & G & & A
\end{array}
$$