Notes de Cours MTH 6415

Olivier Sirois

2018-01-01

Contents

0.1 Examen Finale

Locale: C-539.4 → mardi 24 Avril 0930

Chapter 1

Programmation Dynamique

C'est un ensemble d'outils :

- Model
- Algorithmes
- Corpus théorique..

qui sert à résoudre des problèmes de décisions séquentiels.

Problème de décision séquentiels Problème qui prend une suite (avec un ordre spécifique) de décision ou ces décision sont reliées les unes aux autres avec un objectif qui est d'optimiser (min ou max) d'un certain critère de performance.

l'information qui sert à prendre la décision n'est pas toujours disponible. Alors sans avoir la totalité de l'information nécessaire, sa nous amène à faire la différence entre :

- **Problème à boucle ouverte**. Problème ou toute l'info est nécessaire et disponible. On peut faire des choix au départ et en insistant, (((ordinairement))), on va être capable d'interpréter le problème comme étant déterministe.
- **Problème en boucle fermé**. Problèmes ou on découvre au fur et à mesure l'information utile/importantes pour la prise de décision. Dans cette famille, on à les **problème dynamique stochastique**, qui est l'objectif du cours.

l'objectif est souvent une espérance mathématique. La plupart des concepts stochastique sont applicable au problème déterministe avec une simple arrangement. Par contre, en raison des contraintes temporel, on saute directement dans le stochastique.

Politique (policy) Règle de prise de décision, de façon imprécsie, qui décrit pour chaque situation possible (état du système) la meilleur décision à prendre pour optimiser une fonction objective globale. Cette fonction objective étant souvent une espérance mathématique.

Souvent, on essaie de charactérisé les propriété de la politique optimale. on formule ensuite les modeles d'une certaine façon, on essaie après de les résoudre et finalement on caractérise les solutions.

1.1 Modèle de programation dynamique probabiliste (Processus de décision Markoviens)

Normalement, nous avons:

- Un processus de décision séquentiel qui est découpé dans ce que nous appellons des étapes (N étapes). Habituellement numérotée de O à $N-1$.
- À l'étape x_k, on observe les caractéristiques du système et on prend une décision $u_k \in U_k(x_k)$
- Nous avons une variable aléatoire que nous appelons ω_k qui est généré selon une loi de probabilité $P_k(*|x_k, u_k)$. Cette loi de probabilité peut dépendre de k, x_k, u_k. Mais ce qui est important c'est qu'elle ne dépend pas des valeurs précédentes, c-à-d, qu'elle ne dépend pas de x_m, u_m, ω_m pour $m < k$.

Normalement, on observe ω_k, on paye un cout $g_k(x_k, u_k, \omega_k)$. et à l'état prochaine, x_{k+1} ets donnée par une fonction $f_k(x_k, u_k, \omega_k)$.

Le cout totale: $g_n(x_n) + \sum_{k=0}^{N-1} g_k(x_k, u_k, \omega_k)$. il y a l'état finale x_n, sont cout associé g_n ainsi que les cout de tous les états précédent (la somme sur 0 à N-1).

Exemples:

- Gestion d'inventaire
- Gestion de réservoire (production hydroélectrique)
- autres.. (Deep Q Learning??)

Une politique admissible est une suite de **fonctions** $\pi = (\mu_1, \mu_2...)$ tel que $\mu_k : X_k \to U_k$ ou $\mu_k(x_k) \in U_k(x)$ pour tout $x \in X_k$ et k = 0,.... N-1. **Pour chaque états, j'ai une fonction qui nous donne la bonne décision à prendre**.

Pour résumé:

- X_k = Ensemble des états:
- $U_k(x)$ = ensemble des décision admissible dans l'état x
- D_k = ensemble de perturbations ω_k
- G_k = fonctions de couts
- f_k = fonction de transisition
- x_k = état à l'étape k
- u_k = décision prise à l'étape k
- ω_k = perturbation aléatoire à l'étape k

Bellman Les équations de Bellman (voir INF8215). Le principe d'optimalité est la base de la programmation dynamique.

Pour $0 \leq k \leq N-1$ et $x \in x_k$, posons:

> $J_{\pi,k}$ est le cout espéré total de l'étape k à la fin. Si on est dans l'état x à l'étape k et si on utilise la politique π.
>
> $J_{\pi,k} = E_{\pi,k}[g_n(x_n) + \sum_{m}^{N-1} g_n(x_m, u_m, \omega_m)]$ ou E est l'espérance lorsque $x_k = x$, $u_m = \mu_m(x_m)$ pour m = k,... N-1.

Pour une politique π donnée, on a une équation de récurrence $J_{\pi,M}(x) = g_N(x)$ pour $\forall x \in X_N$

$J_{\pi,k}(x) = E[g_k(x, \mu_k(x), \omega_k) + E_{\pi,K+1}[J_{K+1}(f_k(x, \mu_k, \omega_k))]]$ pour $0 \leq k \leq N-1$ et $x \forall X_k$

ou LHS est le cout immédiat et RHS est le cout future.

typiquement, on cherche une politique π qui va minimiser/maximiser $J_{\pi,0}$, l'espérance de la somme des couts de l'étape 0 jusqu'à l'étape N.

Notons que $\pi^* = (\mu_0^*, \mu_1^*, ...)$ un telle politique optimale. Posons maintenant $J_k^*(x)$ étant le cout espéré total de l'étape k à la fin si on est dans l'état x à l'étape k et donc en fait $J^*(x) = \min_\pi J_{\pi,k}(x)$.

Théorème

1. On a $J_k^* = J_k$ ou les fonctions J_k sont défini par les équations de récurrences (équations de bellman).
2. $J_N(x) = g_N(x), \forall x \in X_N$

3. $J_k(x) = \inf_{u \in U_k(x)} E[g_k(x, \mu_k, \omega_k) + J_{k+1}(f_k(x, \mu_k, \omega_k))]$ pour $0 \leq k \leq N-1$ et $x \in X_k$ et ou l'espérance E est fais par rapport à ω_k qui suit la loi $P_k(*|x, u)$

4. infimum = plus grande borne inferieur de la fonction. Si quelques chose tend vers l'infini, on est obliger de prendre l'infini quand le problème est fini.

5. une valeur de u qui fait atteindre l'infimum est une décision optimale à prendre lorsqu'on est dans l'état x à l'étape k.

6. Donc, on peut définir une politique optimal π^* (dans le cas ou elle existe) par la relation: $\mu_k^*(x) \in arg\min_{u \in U_k(x)} + E[..voirplushaut...]$ et on va voir que $J_k = J_{\pi,k}(x)$ pour tout k et x.

On peut résoudre les équations de récurrence et calculer en même temps une politique optimale par ce qu'on appelle **chaînage arrière**. qui est fait en calculant $J_N(x)$ pour tout $x \in X_N$, pour calculer $J_{N-1}(x) et \mu_{N-1}^*(x), \forall x \in X_{N-1}$ puis ensuite $J_{N-2}, \mu_{N-2}^*...$ ainsi de suite

Les valeurs optimales rechercher est $J_0(x_0)$ ou x_0 est l'etat initiale.

Comment formuler un problème de programmation dynamique Stochastique à horizon fini :

- On spécifie les étapes
- On spécifie les états, les ensemble $x_k, k = 0...N-1$
- On spécifie les décision et les ensemble $X_k(x)$
- On spécifie aussi les perturbations aléatoires ω_k et leurs distributions
- Donner les fonctions g_n et g_k
- Donner les fonctions de transitions f_k
- Écrire les équations de Bellman.

Pour résumer, il faut formuler le problème pour qu'on soit en mesure d'écrire les équations de Bellman.

Principe d'optimalité de Bellman Le principe de Bellman nous dit que si $\mu^* = [\mu_1^*, \mu_2^*, ...]$ est une politique optimale pour notre problème intiales et si on considère un cas $0 \leq k \leq N$ alors la politique tronqués π_k^* est défini comme étant la politique formé par μ^* est une politique optimal pour le problème qui consiste à minimiser l'espérance par rapport à $E_{\mu_k, \mu_k+1}[g_n(x_n) + \sum_{n=k}^{N-1} g_n(x_n, u_n, \omega_n)|x_k]$ pour rapport à μ de l'espérance.

Exemple: Taille d'un lot à fabriquer Un entreprise doit fabriquer m exemplaires d'une certaines pièce pour remplir une commande. Les critères de qualités sont très élevés. Tellement que la compagnie estime que la probabilité qu'une pièce sera acceptable est de p. Les pièces sont fabriquer en lots. Pour fabriquer un lot de taille u, le cout encouru est de $C + cu$. ou C est le cout d'installation et c est le cout par unité. Dans un lot de taille u, le nombre Y de pièce acceptable est une variable aléatoire binomiale. ou $P[Y = y] = (u, y)p^y(1 - p)^{u-y}y = 0, ...u$.

en pratique, on va avoir tendence à fabriquer des lots de taille > M Parce qu'il va presque certainement avoir des rejets. Le nombres de piêces acceptable est quand même inférieur à M, il faudra produire un autre lot pour compenser pour le nombre de piêces manquantes. Supposons qu'on ait assez de temps pour produire N lots. Si on ne remplie pas la commande après N lots, on doit payer une énorme pénalité W.

- Étapes: Lots
- Var d'état x_k = nombre de pièces encore requises avant de produire le lot K+1
- u_k = nombre de pièces produite dans le lot K+1
- y_k = le nombre de bonnes piêces produite dans le lot
- $J_k(x)$ = coût espérer minimal à partir de maintenant si on a K lot de produits et qu'il manque encore x piêces.
- On cherche le cout total espérer. J_0 de M
- On cherche aussi une politique optimale.

Équations de Bellman:

Pour tout k et $x \leq 0$, $J_k(x) = 0$, pour $x > 0$, on a $J_N(x) = W$, $J_k(x) = \min_{u>0}(C + cu + \sum_{y=0}^{u}(u, y)p^y(1 - p)^{u-y}J_{k+1}(x - y)$

Chapter 2

Cours #2

Que faire si les hypothèses de bases sont pas tous respectées?

Dépendance entre les étapes?

- Modèle de planification hydro-électrique
  - Modele de planification hydro-électrique ont souvent des problèmes de 'persistances hydrologique'. Un aport dans les réservoirs dépendent des chutes de pluies de la période courantes (ω_k).
  - chutes de pluie des périodes antérieures. (supposons une seule période)

On va être obliger d'augmenter l'état (rajouter de l'information). l'état doit condenser toute l'info pertinentes sur l'évolution du système dans le passé. On doit rajotuer des information supplémentaires dans l'état. Par exemple, si on avait

- x_k comme étant le niveau du réservoir

On devra considérer des paires:

- $x_k = (y_k, \omega_{k-1})$
- ou x est l'état pour la période k et y représente les niveaux antérieures.

Quand on écrit $f_k(x_k, y_k, \omega_k)$, on va devoir écrire la fonction de transition pour les deux composantes. Dans notre exemple, la deuxiemee composante est triviale. On aura donc :

$$f((y_{k-1}, \omega_{k-1}), u_k, \omega_k) = (y_{k+1}, \omega_k)$$
avec
$$y_{k+1} = y_k + \alpha_1 \omega_{k+1} + \alpha_2 \omega_k - u_k$$

On a toute la lattitude qu'on veut tant qu'on est capable d'écrire les équations. C'est sure que dans un processus avec plusieurs augmentations, on aura besoin d'absorber un grand vecteur aléatoire. (Variable de persistence pour chaque bassin)

On essaie de rester parsimonieux dans la définition de l'état.

2.1 Augmentation de l'état

Référence : Chapitre 4 section 4.2, 4.4

4.3 est une section très intéressante, on aurait couvert le matériel si on avait eu plus de temps.

Supposons qu'à chaque étape k on doit décider si on arrête le système et à ce moment la encaisser un revenue ou un cout ou si on continue (gambler, changer d'emploi, relation..). À partir du moment ou on ce met à réfléchire, on remarque dque l'espace X_k est partitionné en deux:

- le sous-espace des états ou on arrète
- et celui on on continue.

Exemple: vente d'un actif

durant chaque période k, $(0 \leq k \leq N-1)$ on reçoit une offre ω_k que l'on peut accepter à la fin de la période (au temps k+1) si aucune autre offre a été accepter au préalable.

On suppose qu'ω_k sont des variables aléatoires indépendantes et dont la distribution est connues. elles sont indépendantes entre-elles et de nos décisions. Au niveau des décisions, nous allons poser $u_k = 1$ si on vend à la période k et 0 sinon. L'état du système à la période k+1 est :

$$x_{k+1} = [\omega_k, \text{ si } u_0 = u_1 .... = u_k] \ [\Delta, \text{ si on à déja vendu}]$$

Pour des raisons techniques, on suppose à l'origine qu'on a pas d'offre et que nous faisons rien ($x_0 = 0, u_0 = 0$). Les décision admissibles pour k = 1,..., N-1; sont :

- $u_k = 0$ si $x_k = \Delta$
- $u_k \in \{0, 1\}$ si $x_k \neq \Delta$

Pour k=N, $u_N = 1$ si $x_N \neq \Delta$

Une chose un peu particulière est que si $u_k = 1$, on reçoit $x_k = \omega_{k-1}$ ai temps k, on place cette somme au taux d'intérèt r pour les (N-k) périodes restantes. Le principe est que d'avoir l'argent plus tôt nous permet de le réinvestire.. etc, pour laisser savoir que d'avoir une somme d'argent plus tôt est plus avantageux que de l'avoir plus tard. Ce que l'on va dire est que le revenu à l'état k actualisé au temps N, est donc :

$$g_k(x_k, u_k, \omega_k) = [(1+r)^{N-k} x_k, \text{ si } u_k = 1]\ [0, \text{ si } u_k = 0]$$

Soit $J_k(x_k)$ le revenu espéré optimal à partir du temps k jusqu'à la fin, actualisé au temps N. On va obtient:

$$J_k(x_k) = [0, \text{ si } x_k = \Delta]\ [x_N, \text{ si } k = N \text{ et } x_N \neq \Delta]\ [max(1+r)^{N-k} x_k, E_{\omega_k}[J_{k+1}(\omega_k)]]$$

On le premier terme est de vendre tout de suite, le deuxième est d'attendre. La décision, si on à encore le choix, est de vendre (accepter l'offre $\omega_k = x_k$) si et seulement si $x_k \geq \alpha_k$ ou alpha est défini comme étant $\alpha_k = \frac{E_{\omega_k}[J_{k+1}(\omega_k)]}{(1+r)^{N-k}}$

La politique optimale est complètement déterminé par ces seuils la $(\alpha_1, \alpha_2, \alpha_{N-1}....)$

On pose premièrement que $\alpha_N = 0$ pour $x_k \neq \Delta$ car $J_N = 0$ étant donné que c'est la dernière offre.

$V_k(x_k) = \frac{J_k(x_k)}{(1+r)^{N-k}} = [x_N, \text{ si } k = N]\ [max[x_k, \frac{E_{\omega_k}[J_{k+1}(\omega_k)]}{(1+r)^{N-k}}]]$ si $k < N$

ou le deuxième terme est $\frac{E_{\omega_k}[V_{k+1}(\omega_k)]}{(1+r)}$
$V_k(x_k) = max[x_k, \alpha_k]$

on a $\alpha_k = E_{\omega_k}[V_{k+1}(\omega_k)]$, $(1+r)\alpha_k = E_{\omega_k}[V_{k+1}(\omega_k)] = E_{\omega_k}[max(\omega_k, \alpha_{k+1})]$

Si on regarde la fonction cumulative d'ω_k ...

$$(1+r)\alpha_k = E[\omega_k\ I[\omega_k > \alpha_{k+1}]] + E[\alpha_{k+1}\ I[\omega_k \leq \alpha_{k+1}]] = E[\omega_k\ I[\omega_k > \alpha_{k+1}]] + \alpha_{k+1} P[\omega_k \leq \alpha_{k+1}]$$

Cela nous permet de calculer les α_k par récurrence. On a $\alpha_N = 0$,

$$\alpha_{N-1} = \frac{E[\omega_{N-1}]}{(1+r)},$$

Proposition:

Si les ω_k i.i.d, alors $\alpha_k \geq \alpha_{k+1} \forall k$:
Démonstration:
On note par ω une variable aléatoire qui à la même distribution que les ω_k. Il suffit de démontrer que $V_k \geq V_{k+1}$ pour x non-négatif et $1 \leq k \leq N-1$. Par

récurrence sur k.

Pour k= N-1, on a $V_k(x) \geq x$ et $x = V_N(x)$, Si on suppose que $V_{k+1}(x) \geq V_{k+2}(x), \forall x \geq 0$, alors:

$V_k(x) = max(x, \frac{E[V_{k+1}(\omega)]}{(1+r)})$, $\geq max(x, V_{k+1}(x))$

D'où $V_k(x) \geq V_{k+1}(x)$

Que se passe-t-il si $N \to \infty$? Ou de façon équivalente, si $k \to -\infty$?

Si on peut borner la suite des α_k, cela montrera que cette suite converge quand $k \to \infty$. On avait $(1+r)\alpha_k = E[\omega_k * I[\omega_k > \alpha_{k+1}]] + \alpha_{k+1}P[\omega_k \leq \alpha_{k+1}]$. on peut remplacer tout le tralala par $E[\omega] + \alpha_{k+1}$, On en tire:

$\alpha_k \leq \frac{E[\omega]+\alpha_{k+1}}{(1+r)} = \frac{E[\omega]}{(1+r)} + \frac{E[\omega]+\alpha_{k+2}}{(1+r)^2} = E[\omega]\frac{1+r}{r} < \infty.$

Les résultats sont bornés par quelques choses qui dépendent de l'espérence d'ω. On est capable de montrer que lorsque $k \to \infty, \alpha_k \to \alpha^-$ une constante qui satisfait :

$$(1+r)\alpha^- = E[max(\omega, \alpha^-)] = E[\omega * I[\omega > \alpha^-]] + \alpha^- P[\omega \leq \alpha^-]$$

La politique stationnaire optimale est définie par un seuil unique α^-.

2.2 Modèle symmétrique : Achat avant une date limite

On veut acheter le moins chère possible une actif avant une date limite. NOTE: ACHETER UNE VOITURE LE 30 JUIN, ou la fin de chaque trimestre

Cas exactement symmétrique, sauf que le but c'est de minimiser au lieu de maximiser. Nous allons acheter si $x_k < \alpha_k$ ou les alphas k sont croissants.. etc.

Dans le cas ou les prix sont corrélés (billets d'avion). On suppose que $x_{k+1} = \omega_k = \lambda x_k + \xi_k, 0 \leq k \leq N-1, \lambda \in (0,1)$ est une constante et les ξ_k sont des variables aléatoires.

2.3 Ensemble D'arrêt Absorbant

et Règle de 'un coup à l'avance'

On considère un modèle stationnaire générale dans lequel la décision u_k à l'étape k implique de payer un coût terminale $t(x_k)$ si on arrête, ou de continuer À l'étape N si on n'a pas déjà arrêté, on doit le faire et payer $t(x_k)$.

- $J_N(x_N) = t(x_N)$

- $J_k(x_k) = min(t(x_k), min_{u \in U(x_k)}(E_{\omega_k}[g(x_k, u_k, \omega_k)] + J_{k+1}[f(x_k, u, \omega_k)]))$

ou (Modèle Stationnaire):

- $f_k = f, \forall k$
- $g_k = g, \forall k$
- $U_k = U, \forall k$

pour $k < N$, il est optimale de s'arrêter au temps $k < N$, si et seulement si $x_k \in T_k$ ou T_k est l'ensemble des X tel que:

$t(x) \leq min_{u \in U(x)} E[g(x, u, \omega) + J_{k_1}(f(x, u, \omega))]$

On voit qu'on peut montrer que $J_{N-1}(x) \leq J_N(x)$ On fait juste comparer les fonctions de valuers aux deux étapes. Si on est dans l'état, on à obligatoirement cette relation, en simple raison de notre définition.

$J_k(x) \leq J_{k+1}$

Il découle que $T_k \subseteq T_{k+1}$ pour $0 \leq k \leq N - 1$ "T de k plus 1 contient T "

Proposition:

Si l'ensemble T_{N-1} est absorbant, tant que l'on ne s'arrête pas, autrement dit, le fait d'avoir x appartenant à T_{N-1} implique que la fonction de transition f appartient à T_{N-1}. $= f(x, u, \omega) \in T_{N-1}$

alors: $T_k = T_{N-1}$ pour tout k ¡ N.

Chapter 3

21 Mars

Si l'ensemble T_{n_1} est absorbant tant qu'on ne s'arrète pas, autrement dit, si $x \in T_{n_1} \rightarrow f(x,u,\omega) \in T_{n_1}$ alors $T_{n-1} = T_k, \forall k < N$. Si $x_{n-2} = x \in T_{n-1}$, alors $x_{N-1} = f(x_{N-2}, u, \omega) \in T_{N-1}$
de sorte que
$J_{N-1}(x_{N-1}) = t(x_{N-1})$
et donc
$t(x) \leq min_{u \in U(x)} E[g(x,u,\omega) + J_N(f(x,u,\omega))] = E[g(x,u,\omega) + t(f(x,u,\omega))]$

$t(x) \leq min_{u \in U(x)} E[g(x,u,\omega) + J_{N-1}(f(x,u,\omega))]$

$\rightarrow x \in T_{N-2}$

On avait mentionné que $T_k \in T_{k+1}, \forall k \in [0,N]$

Autrement dit, on a $T_{N-1} = T_{N-2}$

En répétant cette preuve pour k = N-3,... 0, on tire la conclusion que:

$T_k = T_{\omega-1}, k = 0, ..., N-1$
si T_{N-1} est absorbant, alors la politique optimale à chaque étape est de s'arrêter si et seulement si il est préférable de s'arrêter maintenant que de continuer et de s'arrêter obligatoirement à la prochaine étape. Autrement dit, il suffit de regarder un coup à l'avance. Ce qu'on appelle en anglais *One step look ahead policy.*

3.0.1 Exemple

Un voleur qui sait calculer..

À chaque période k (chaque nuit), un voleur décide peut tenter un nouveau vol ou prendre sa retraite avec son profit déjà accumuler x_k. S'il tente un vol,

il perd tout le profit qu'il à fait précédement de plus que d'aller en prison (il ne peut plus faire d'autre vol) avec une probabilité p. Il a une probabilité $(1-p)$ de faire un gain ω_k qui est aléatoire.

Évidemment, notre problème est stationnaire. Après N période, il doit nécessairement prendre sa retraite étant donné qu'il est trop vieux pour voler avec son profit accumuler x_N s'il ne l'a pas fait avant et s'il n'a pas été attrapé. Il veut évidemment maximiser son profit espéré total $E[x_N]$

Les équations de Bellmans sont:

$$J_N(x_N) = x_N$$
$$J_k(x_k) = max(x_k, (1-p) * E[J_{k+1}(x_k + \omega_k)] + p * 0)$$
$$\text{On a ici : } T_{N-1} = x \| x \geq (1-p)(x + E[\omega]) \cup \Delta$$
$$= x \| x \geq E[\omega]\frac{(1-p)}{p} \cup \Delta$$
$$\text{Ou } \Delta \text{ est l'arrestation } (=0)$$

On voit clairement que T_{N-1} est absorbant au sens de la proposition puisque si j'ai $x \in T_{N-1}$, deux cas possibles, C'est soit x = {Δ et alors $f(x, u, \omega) = \Delta$, sinon, $x = f(x, u, \omega) = x + E[\omega]$

En conclusion, il est optimal de s'arrêter dès que $x_k \geq E[\omega]\frac{1-p}{p}$ peu importe k. Dès que le profit accumuluer dépasse ce ratio, il est optimal d'arrêter.

3.1 Ordonnancement

3.1.1 Argument d'échange des voisins

On a un ensemble de N tâches à faire (des processes) à exécuter. On veut miniser un certain critères de performances qui s'exprime comme l'espéance de la somme des coûtspour les différentes tâches. Les modèls considérés sont stochastique mais l'information obtenus au cours des premières étapes n'est pas utiles pour améliorer les décisions futures. De sorte que la politique optimales est en boucle ouverte (on peut trouver l'ordonnancement optimale à priori). **l'argument d'échange des voisins** dit que:

$L = \{i_0, i_1, ...i_i, i_j, ...i_{N-1}, i_N\}$ est un ordonnancement optimale. On peut avoir un ordonnance non-optimale L' tel que:

$L' = \{i_0...i_j, i_i, ...i_N\}$ ou le cout total esperé de L ne doit pas dépasser le cout L' sous aucuns circonstances. En générale, ceci ne donne que des conditions nécessaire d'optimalité, mais parfois, il devient évident que ces conditions sont aussi suffisantes.

Exemple

Ordonnancement des questions d'un quiz (voir bertsekas)

Il y a N questions auxquelles on peut répondre dans l'ordre que l'on veut. On répondra correctement à la question i avec probabilité, et si on le fait, on va gagner R_i. Dès que l'on échoue à une question, on doit arrêter le quiz et conserver le nombre de points que nous avons accumuler jusqu'à présent. l'objectif est de maximiser l'espérance de points accumuler durant l'examen. On note que la politiuqe optimale est en boucle ouverte car une fois que nous avons répondu aux k premières questions, on n'aura pas plus d'information qu'au départ pour changer l'ordonnancement des N-k questions suivantes. Soit $J_(S)$ le revenu espéré pour une suite ordonné de questions S.

Avec les définitions précédentes de L et L', où L est optimale, on a :

$L = \{i_0, i_1, ...i_i, i_j, ...i_N\}$ et $L' = \{i_0, ...i_j, i_i, ...i_N\}$
$J(L) = J(\{i_0, ...i_{k-1}\}) + p_{i_0}p_{i_1}...p_{i_k}(p_iR_i + p_ip_jR_j)..+p_0...p_ip_jJ(\{i_{k+2...i_{N-1}}\})$

$J(L')$ est pareil, sauf à l'endroit ou la séquence change (i,j), les termes vont être similaire mais avec les lettre i et j inversé.

On doit par contre avoir $J(L) \geq J(L')$ alors $J(L) - J(L') \geq 0$ donc:

Sous l'hypothèse que $p_i, \forall i \in [0, k] > 0$, on en tire que $p_iR_i + p_ip_jR_j > p_jR_j + p_jp_iR_i$ on peut simplifier:

$p_i(1 - p_j)R_i \geq p_j(1 - p_i)R_j$

$\frac{p_iR_i}{1-p_i} \geq \frac{p_jR_j}{1-p_j}$

On remarque alors que cette condition suffit pour déterminer L. On doit simplement trier les équations en ordre décroissant de ratio $\frac{p_iR_i}{1-p_i}$.

Boucle ouverte = trouver la politique optimale dès le début.

Dans le dernier devoir, nous allons devoir résoudre un problème de ce genre.

3.2 Corrigé Dev. 3

- Étapes : Invitations

  étapes k si on a déjà fait k invitations.

- 3 étapes + 1

- Posons C = {Anna, Béatrice, Clara}
- état x_k le sous ensemble de C qui n'ont pas été invité jusqu'au début de l'étape k
- Décision: u_k personne que l'ont invite à l'étape k.
- $U(x_k)$ est l'ensemble des décision $u \in x_k$
- Probabilité que la personne u accepte l'invitation: $p(u_k)$
- $V_c, c \in C$ utilié associés à la camarade c
- $J_k(x)$: utilité esperée optimale si on a pas de cavalière au début de l'étape k et que l'ensemble des personne non-invités jusqu'à la est X

On peut alors écrire les équations de Bellman suivantes :

$$J_3(\{\}) = 0$$
Pour k = 0,1,2... on a:
$$J_k(x) max_{u \in X} [p_{uk} v_u + (1 - p_{uk}) J_{k+1}(x_{k+1})]$$

En principe, on fait des tableaux pour chaque étapes représentant les valeurs d'utilité pour chaque $u \in U(x)$.

Chapter 4

Cours 4 avril

7.16 : la rigueur peut faire déraper..

Itération de valeur peut être mieux aprèter. On explore difficilemenet l'espace des politiques

4.1 Itération Politique

Le but est d'explorer l'étendu des espaces de politique. C'est un algorithme dans lequel on commence par **choisir une politique stationnaire** μ, qui va être notre première approximation de la politique optimale. Ensuite, nous allons répéter une boucle dans lequel nous essayerons d'améliorer cette première politique.

Pour faire sa, on calcul un J tel que $J = T_\mu J$. Nous allons ensuite optimiser la politique tel que $T_\mu J = T(J)$. Après chaque itération de notre algorithme, nous allons avoir une nouvelle politique qui, si tout est bien fait, va converger vers $\mu*$.

Tant que $|J - T(J)|$ est trop grand, nous allons retourner μ.

4.1.1 Proposition (s)

À chaque itération de l'algorithme, la fonction J ne peut augmenter en aucun point par rapport au J précédent (il faut que sa s'améliore). Elle ne peut que diminuer. De plus, lorsque μ ou J ne change pas par rapport à l'itération précédent, nous avons atteint un minimum local de la politique $\mu*$.

Dans le cas ou le nombre de politique stationnaires est fini, c-à-d, que les ensemble X et U sont finies, on peut remplacer 'trop grand' par 'plus grand que

0'. Et l'algorithme va toujours s'arrêter après un nombre finis d'itérations et va nous procurer une solution optimale. (On essaye toute les politique et on prend la meilleure..).

Soit U la politique à une itération donnée et μ la politique à l'itération suivante. On veut montrer que $J_\mu \leq J(U)$. On a:

$T_\mu(J_U) = T(J_U) \leq T_U(J_U) = J_U$

Par la monotonicité de T_μ, $J_U = lim_{k\to\infty}(T_\mu^k(J(U)) \leq J(U)$

$T_\mu \leq J_U$
$T_\mu^2(J_U) \leq T_\mu(J_\mu(U) \leq J\mu$

Qui implique que $T^k < T^{k-1} < T^{k-2}... \leq J_\mu..$

à la limite de l'optimisation, nous allons avoir:

$J_\mu = T_\mu J(U) = T(J_U)$

et donc $J_U = J*$, car c'est le seul point fixe de $J*$. Si μ n'est pas encore optimale, nous allons toujours diminuer le prochain T, par contre, lorsque nous n'allons plus pouvoir le diminuer, nous avons avoir atteint la limite de l'optimisation. Donc si je vois une nouvelle politique, à chaque itération, sa veut dire que je ne pourrai pas faire plus d'itération que le nombres total de politique stationnaires.

On a $|X| = 3$ et $|U| = 2$. Il y a au total $|U|^{|X|} = 2^3 = 8$ politique possibles, alors nous avons tous simplement besoin de voir la meilleur politique des 8, nombre d'itérations possible = 8...

La limite de cette méthode est lorsque l'ensemble des états est très grand. Sa peut être très couteux de rouler l'algorithme lorsque l'ensemble des états possible est très grand... On peut voir que le but de l'algorithme est tout simplement d'itérer chaque politique possible et de voir la meilleur.

Dans ce cas, on peut utiliser **l'algorithme d'approximation successive** pour un certain nombre d'itération. C'est un algorithme hybride dans lequel on va remplacer 'Trouver J..' par 'J $\leftarrow T_\mu^k(J)$. Si le nombre d'itération est positif, la suite des fonctions J visitées va converger vers $J*$, dans le sens que la norme $|J - J*| \to 0$. Si jamais le nombre de politique stationnaire est fini, nous allons quand même nécessairement converger vers la politique optimale dans le même nombre d'itération que l'algorithme d'itérations de politiques.

4.1.2 Lien avec la programmation linéaire

Si $J \leq J*$, alors $J \leq T(J)$. En fait, on peut dire que $J*$ est le plus grand J tel que $J \leq T(J)$. Si $|x| = m$, alors le vecteur $J*$ doit être la solution optimale du programme linéaire. on cherche à maximiser $\sum_{i=1}^{n} J(n)$, donc $J(i) \leq g(i,u) + \alpha(P_{i1}(u)J(1) + P_{i2}(u)J(2) + P_{i3}(u)J(3)...)$.

P est la probabilité de passer à l'état i à l'état j à partir de u. Cette relation la doit être vraie pour $i = 1, ...n$ et $u \in U(i)$.

Ce P.L possède n variables représentant chacun des états et $|U(1)|+|U(2)|+|U(3)|+...|U(n)|$ contraintes.
Résolu habituellement par une méthode duale...

Chaque politique μ correspond à une base du PL duale. Alors on peut faire une relation entre l'exploration des politique dans l'algorithme et l'exploration des bases du duales dans la résolution d'un programme linéaire par méthode du simplex.

4.1.3 Cout moyens par étapes

Considérons le modèle avec $X = \{1, ..., n\} et \alpha \leq 1$. Pour le modèle actualisé avec $\alpha < 1$, notons $J_{\alpha,\mu}$ et $J_\alpha*$ les valeurs de J_μ et $J*$ donnée. On va faire tendre α vers 1.

Pour une politique stationnaire μ, la $lim_{N\to\infty} \frac{1}{N} E[\sum_{k=0}^{N-1} g(x_k, \mu(x_k))]$.

$= lim_{N\to\infty} lim_{\alpha\to 1} \frac{E[\sum_{k=0}^{N-1} \alpha^k g(x_k,\mu(x_k))]}{\sum_{k=0}^{N-1} \alpha^k}$

On effectue value iterations jusq'à temps que la valeur converge. Par contre, en utilisant $\alpha = 1$, nous pouvons prendre la moyenne des valeurs comme étant la moyenne du cout par étapes.

$= lim_{\alpha\to 1} \frac{J_{\alpha,\mu}(x)}{\frac{1}{1-\alpha}}$. On choisit un état, disons t, comme étape de référence et on pose:

$h_{\alpha,\mu}(i) = J_\alpha(i) - J_{\alpha,\mu}(t)$
$\lambda_{\alpha,\mu} = (1-\alpha)J_{\alpha,\mu}(t)$
$h_\mu(i) = lim_{\alpha\to\infty} h_{a,u}(i)$
et
$\lambda_u = lim_{\alpha\to 1} \lambda_{\alpha,\mu}$

$h_{\alpha,\mu}$ et $h_\mu(i)$ représente un cout différentiel par rapport à l'état de référence. λ_μ représente **le cout moyen par étapes** sur horizon infini sous la politique μ. Si c'est possible de passer de n'importe quel état à n'importe quel autre avec une probabilité p (au sens markovien), ce cout ne dépendra pas de x mais aussi de p.

.....

P_μ est la matrice de transistion associés à la politique μ. Qui représente la probabilité de passer d'un état i à un état j selon la politique μ. Dans le modèle en coût actualisé, on peut ré-écrire l'équation de Bellman associé à la politique μ:

$J_\mu = g_\mu + \alpha * P_\mu(J_\mu)$.

en tenant compte d'α, on peut écrire :

$J_{\alpha,\mu} = g_\mu + \alpha P_\mu J_{\alpha,\mu}$

$h_{\alpha,\mu} + J_{\alpha,\mu}(t) = g_\mu + \alpha P_\mu(h_{\alpha,\mu} + J_{\alpha,\mu}(t))$ pour l'état t..

$\lambda_{\alpha,\mu} + h_{\alpha,\mu} = g_\mu + \alpha P_\mu h_{\alpha,\mu}$

Lorque $\alpha \to 1$, nous avons $\lambda_\mu + h_\mu = g_\mu + P_\mu h_\mu = T_\mu(h_\mu)$

Pour les valeurs optimales, nous avons $h_\alpha * (i) = J *_\alpha (i) - J *_\alpha (t)$, $\lambda_\alpha = (1-\alpha)J *_\alpha (t)$

$h * (i) = lim_{\alpha\to 1} h_\alpha * (t)$
$\lambda * = lim..\lambda *_\alpha$

h est le cout différentiel par étapes, et lambda est le cout moyen par étapes sur horizon infini, qui est ultimememnt ce qu'on cherche. L'équation de Bellman se réécrit comme:

$\lambda_\alpha + h *_\alpha = min_\mu[g_\mu + \alpha P_\mu h *]$

et devient à la limite:

$\lambda * + h * = min_\mu[g_\mu + P_\mu h_\mu] = T(h*)$

proposition

Il existe une constante λ et une fonction $h \in B(x)$ tel que $\lambda + h = T(h)$ si et seulement si $\lambda = J * (i) = min_\pi J_\pi(i)$ pour tout $i \in X$ Si $T_\mu(h)$ pour ce h, alors la politique μ est optimale.

De même, il existe $\lambda_\mu eth_\mu \in B(x)$ tels que

$\lambda_\mu + h_\mu = T(h_\mu)$

si et seulement si $\lambda_\mu = J_\mu(i), \forall i \in X$

4.2 Cours 11 avril

Warren&Powell

4.2.1 devoir 4 et 5

1.14 dev 4 Équations de Bellman..
$J_k(x_k) = max_{u\in[0,1]}[(1-u)x_k + E_{\omega_k}[J_{k+1}(x_k+\omega_k)ux_K]]$

$\omega_k \to \omega_k \geq 0, E[\omega_k] = \bar{\omega} > 0$

On peut montrer par récurrence que $J_k(x_k)$ est de la forme $\alpha_k x_k$ avec $\alpha_k = [1+\alpha_k$, si alpha k+1 omegabar $\leq 1][(1+\bar{\omega})\alpha_k + 1]$si alpha k+1 ¿ 1.

On remarque que $J_k(x_k)$ est linéaire en u $\to u_K* = 0$ ou 1

On examine ensuite les cas par un.

- $\bar{\omega} > 1$ On démontre facilmenet par récurrence que $u_k* = 0, k = 0...N-1$ et $J_k(x_k) = (N+1-k)x_0$
- $0 < \bar{\omega} < 1/N$ très facile, on démontre par récurrence que $u_k* = 1, ...k = 0, ..., N-1...$ $J_k(x_k) = (1+\bar{\omega})^{N-k}x_k$
- On montre que pour $\bar{k}$ tel que $\frac{1}{k} < \bar{\omega} < \frac{1}{k}$, on à:

  $k \geq N - \bar{k}$, on à $\alpha_{k+1}\bar{\omega} \leq 1$ et $u_k* = 0$, et 1 pour le cas contraire

4.19 état:
$x_k = 1$ si stationné, 0 sinon

$\omega_k = 1$ si c'est libre, 0 sinon

$u_k(1) \in [0]$, si on est déjà stationné, on ne fais rien

$u_k(0) \in [0,1]$, si on est pas stationné, on a l'option de rien faire ou de se stationné à l'emplacement k

$f_k(1, \omega_k, 0) = x_{k-1} = 1$
$f_k(0, \omega_k, 0) = x_{k-1} = 0$
$f_k(0, 0, 1) = x_k = 0$
$f_k(0, 1, 1) = 1$
$f_k(0, \omega_k, u_k) = min(\omega_k, u_k)$

$J_k(x_k)$ =coût espéré si on est dans l'état x_k quand on arrive à la place k
$J_0(1) = C$
On a $J_k(0) = 0, k = 0, ...,$

$J_k(1) = min[J_{k-1}(1), p(J_{k-1} + k) + (1-p)J_{k-1}(1)]$

$F_k = min[F_{k-1}, pk + (1-p)F_{k-1}]$

$= p * min[k, F_{k-1}] + (1-p)F_{k-1}$

avec $F_0 = C$

ii) Il est clair que $F_k \leq F_{k-1}, \forall k$, F_k décroit avec k tandis que k augmente. ce qui implique une seule intersection. Les fonctions $f_1(k) = F_k$ et $f_2(k) = k$ doivent se croiser en un certain $k*$.

Soit k*, le plus petit k tel que k ¿ F_k, il est optimal de continuer jusqu'à temps qu'il y a une seule intersection entre k et $F_k = F_{k-1} = ...F_{k*-1}$

Pour k ¡ k*, on à :

$F_k = pk + qF_{k-1}$ et on peut itérer cette relation. On calcule $F_{k*} = \frac{1}{1-q}[(k * -1) - k * q + q^{k*}] + q^{k*-1}C$

On a aussi:

k* > F_{k*-1}

et on peut continuer..

4.23 S = suite de couloirs

J(S) = probabilité que thésée s'échappe en essayant les couloirs de S dans l'ordre, étant donnée qu'il est encore prisonnier/vivant au début de la séquence.

L = $\{l_1, l_2, l_3...\}$

J(L) = $J(\{i_0, i_1, i_2, i_3..\})$ probabilité qu'on s'échappe de la liste..
+ $[\prod_{l=0}^{k-1}(1 - p_l - q_{i_l})](p_i + (1 - p_i - q_i)p_j)$

$+ [\prod_{l=0}^{k-1}(1-p_l-q_{i_l})]((1-p_i-q_i)+(1-p_j-q_j))J(\{i_{k+1},....i_{N-1}\}$
$= p_i + (1-p_i-q_i)p_j \geq p_j + (1-p_j-q_j)p_i$
$= -q_i p_k \geq -q_j p_i = \frac{p_j}{q_j} \leq \frac{p_i}{q_i}$

7.16 États: les n états possibles de la machine $\{1, 2, ...n\}$
Coûts: $g_(1) < g(2) < g(3)... < g(n)$
Remplacement: R
Transitions: on a pour $i < m$, $p_{i(i+1)} > 0$, on peut seulement transitionner à i+1 ou à i.
actions: (N) ne rien faire, c-à-d. qu'on laisse opérer la machine et (R), qui est de réparer la machine à l'état 1 et la laisse la pendant une période. tout sa à un coût R.

Les transitions associés à la décision R sont $p_{ij}(R) = 1, j = 1, i = 1, ...n$ et $0, j \neq 1, i = 1, ...n$
Pour la décision de rien faire, on a bien

$p_{ij}(N) = [> 0$ j=i ou i+1 $][0$j $\neq 1$, $j \neq i+1]$

les coûts associés sont :

- $\bar{g}(i, N) = g(i), i = 1, ...n$
- $\bar{g}(i, R) = R + g(1), i = 1, ..., n$

Le coût actualiser si on début dans l'état i, pour i¡n on a $J*(i) = min[g(i) + \alpha[p_{ii}J*(1) + (1-p_{ii})J*(i+1)], R + g(1) + J*(1)]$

pour l'état terminal n, $J*(n) = min[g(n) + \alpha J*(n), s]$