Biotechnológia és bioinformatika – formai ajánlások

Sorozatoldal

Címoldal

Copyright-oldal
szerző(k)
lektor
10-15 kulcsszó
10 soros összefoglaló

Tartalomjegyzék

0.1. Szekvenciális döntési folyamatok

0.1.1. Optimális döntés

Egy döntési helyzetben tipikus feladat a rendelkezésre álló információ alapján egy, vagy több kritériumnak megfelelően a legjobb lehetőség kiválasztása a felmerülő opciók közül. Ilyen egyszerű helyzet lehet egy üdülés célpontjának kiválasztása, miközben adott a költségkeret vagy a kiindulási hely, és vannak bizonyos preferenciáink pl. tengerpart, pálmafák.

Egy döntéselméleti helyzet leírásához definiálni kell egy rendszer **állapotait** ($s \in S, states$), az egyes állapotokban elérhető **lépéseket** ($a \in A, actions$), és meg kell határozni az egyes állapotok **hasznosságát** ($U(s), utility$). Egyes állapotokból a megfelelő lépés kiválasztásával lehetséges más állapotokba történő átlépés. A hasznosságfüggvény az állapotok felett definiált valós függvény, amely teljesíti a racionális döntéshozóra vonatkozó **hasznossági axiómákat**. Mivel a hasznosságfüggvény valós függvény, ezért az axiómák többségét definícióból fakadóan kielégíti, ezek a sorrendezhetőség, tranzitivitás, folytonosság, monotonitás. A szerencsejátékokra vonatkozó további axiómák a következőek:

- Helyettesíthetőség – Ha egy olyan szerencsejátékot játszunk, ahol p valószínűséggel az egyik, $1-p$ valószínűséggel a másik állapotba jutunk, és van két állapot, melynek hasznossága azonos (s_i, s_j), akkor azok a szerencsejátékban felcserélhetőek:

$$U(s_i) = U(s_j) \Rightarrow \exists p[p, s_i; 1-p, s_k] \sim [p, s_j; 1-p, s_k].$$

- Felbonthatóság – Ha egy olyan összetett szerencsejátékot játszunk, ahol az első szerencsejáték egyik kimenete egy másik szerencsejáték, akkor az ekvivalens egy három kimenetű összetett szerencsejátékkal, függetlenül az állapotok hasznosságától, illetve p és q értékétől:

$$[p, s_i; 1-p, [q, s_j; 1-q, s_k]] \sim [p, s_i; (1-p)q, s_j; (1-p)(1-q), s_k]$$

Legegyszerűbb esetben a lépések eredménye **determinisztikus** (1 ábra), így az optimális döntés minden s állapotban az az a lépés, amely az elérhető legnagyobb hasznosságú állapothoz vezet (1 egyenlet).

$$U(a^*|s) = max_{a \in A} U(a|s). \tag{1}$$

Abban az esetben viszont, amikor a rendszer valamilyen bizonytalanságot tartalmaz, a lépések kimenetele **nemdeterminisztikus** (1 ábra), így egy eloszlást definiálhatunk a lépések kimenete felett: s_i állapotban, a lépés mellett annak valószínűsége, hogy s_j állapotba jutunk $P(s_j|a, s_i)$.

□ □

1. ábra. Bal oldal: Determinisztikus döntési helyzetben a_i választása mellett s_i állapotba jutunk. Jobb oldal: Nemdeterminisztikus döntési helyzetben a választása mellett $P(s_i|a, s)$ valószínűséggel s_i állapotba jutunk.

A racionális döntéshozóra vonatkozó utolsó két axiómából következik, hogy nemdeterminisztikus esetben az optimális döntés mindig azt az a^* lépést jelenti, ami maximalizálja a várható hasznosságot (maximal expected utility, MEU):

$$MEU(a^*|s) \doteq \max_{a \in A} EU(a|s) = \max_{a \in A} \sum_{s_i \in S} U(s_i) P(s_i|a, s), s \in S. \tag{2}$$

0.1.2. Szekvenciális döntés

Az előző fejezetben az **egylépéses**(myopic) esetet vizsgáltuk, bizonyos esetekben azonban előfordulhatnak **szekvenciális** döntési helyzetek (2 ábra). Például egy körutazás esetén minden érintett helyhez rendelhetünk hasznosságot, a pillanatnyi tartózkodási helytől pedig függ a másnap elérhető helyek halmaza. A bizonytalanságot a rendszerben a megbízhatatlan idegenvezető jelentheti.

Egylépéses esetben egy állapot hasznosságát definíció szerint az $U(s)$ hasznosságfüggvény határozta meg. Szekvenciális esetben azonban egy állapot hasznosságát befolyásolja a belőle elérhető további állapotok hasznossága is. Az egyes állapotokra vonatkozó $U(s)$

hasznosságfüggvények felhasználásával a t diszkrét időpontban s állapot hasznossága egy rekurzív képlettel írható le:

$$U^t(s) = U(s) + \sum_{s_i \in S} U^{t+1}(s_i) P(s_i|a, s). \tag{3}$$

Tehát $U(s)$ az a hasznosság, amit s meglátogatása ér, $U^t(s)$ jelenti azt a hasznosságot, amely figyelembe veszi az s állapotból elérhető teljes döntési gráfot, és az abból számolt várhatóértéket. Az egyenlet második tagja egy várhatóérték az s állapotból elérhető állapotok hasznossága felett. A várható maximális hasznosság ennek megfelelően módosul:

$$MEU(a^*|s) = \max_{a \in A} \sum_{s_i \in S} U^t(s_i) P(s_i|a, s), s \in S. \tag{4}$$

2. ábra. Szekvenciális döntés

Feltételezve, hogy minden állapot elérhető minden állapotból, és egy állapotban többször is tartózkodhatunk, a 3 egyenlet számítása n **véges** lépést feltételezve dinamikus programozással $O(n|A||S|)$ ideig tart. Az utolsó lépésben egyszerűen adódik mekkora az egyes állapotok hasznossága ($U(s)$), majd az $n-1$ lépésben a hasznosságok már a 3 képlettel kiszámolhatóak, egészen a kezdő lépésig. Természetesen ez a módszer nem alkalmazható végtelen számú lépés esetén.

Végtelen lépésszám esetén ($n \to \infty$) a 4 egyenlet nem alkalmazható, mert a nyelő csomópontoktól eltekintve a maximális várható hasznosság minden állapotra végtelennek adódik. Egy lehetséges megoldás a jövőbeni jutalmak leszámított értékével történő számítás, ekkor a jövőbeni jutalom egy $0 < \gamma \leq 1$ együtthatóval megszorzott értékével számolunk:

$$MEU'(a^*|s) = \max_{a \in A} \sum_{s_i \in S} \gamma U^t(s_i) P(s_i|a, s), s \in S. \tag{5}$$

Az 5 egyenletet lépésenként kibontva $[a_0^*, a_1^*, \ldots, a_i^*, \ldots]$ optimális akciók egy sorozatát hajtjuk végre. Ha feltételezzük, hogy egy állapotból maximum k másik állapotba lehet eljutni, $\gamma < 1$, az állapotváltások valószínűségének maximuma P_{max} és az $U(s)$ függvény maximuma U_{max}, akkor a következő kifejezésre jutunk:

$$\begin{aligned} MEU'(a_0^*, a_1^*, \ldots, a_i^*, \ldots|s) &\leq U(s) + \gamma k U_{max} P_{max} + \gamma^2 k U_{max} P_{max} + \ldots + \gamma^i k U_{max} P_{max} + \ldots \\ &= U(s) + k U_{max} P_{max} \sum_{i=0}^{\infty} \gamma^i \\ &= U(s) + \frac{k U_{max} P_{max}}{1-\gamma}. \end{aligned} \tag{6}$$

A 6 egyenletből látszik, hogy a $\gamma < 1$ feltétel, és a leszámított számítás esetén a maximális várható hasznosság felülről becsülhető.

Markov döntési folyamatok

Az előzőekben bevezetett döntéselméleti formalizmus meghatározó jellemzője a **Makorv-tulajdonság**: annak valószínűsége, hogy a folyamat t időpillanatban s_i állapotba kerül, csak a folyamat $t-1$ időpillanatban felvett s állapottól függ, amennyiben az ismert ($P(s_i|a,s)$). Tehát s állapot ismeretében a korábbi állapotok ismerete nem szükséges az állapotátmenetek valószínűségének meghatározásához.

A Markov döntési folyamat a szekvenciális döntési folyamatokat leíró, az eddigiektől kicsit eltérő, azonban azzal teljesen ekvivalens igen elterjedt formalizmus. A Markov-döntési folyamat definiálja az S_0 kezdőállapotot, a $T(s,a,s')$ állapotátmenet modellt (transition) és az $R(s)$ jutalomfüggvényt (reward). A $T(s,a,s')$ állapotátmenet függvény ekvivalens a korábban definiált $P(s_i|a,s)$ feltételes valószínűséggel, míg az $R(s)$ jutalomfüggvénynek az $U(s)$ hasznosságfüggvény felel meg. Markov-döntési folyamatokkal kapcsolatban szokás beszélni az úgynevezett eljárásmódról (policy), mely a döntéshozónak minden állapotra meghatározza, hogy adott állapotban melyik lépést válassza. Optimális eljárásmódnak (optimal policy) nevezzük azt az eljárásmódot, amely a maximális várhatóértékű lépést adja. A korábban bevezetett fogalmakkal az optimális eljárásmód nem jelent mást, mint hogy a döntéshozó minden lépésben az a^* lépést választja (véges lépésszám esetén a 4 , végtelen lépésszám esetén a 6 egyenlet alapján).

0.1.3. Az információ értéke

Az előzőekben a szekvenciális döntéseket úgy modelleztük, hogy minden döntési lépés után a rendszer állapotot vált. Azonban sok esetben döntési opció a szekvenciális döntési sorozatból történő kilépés, a megállás. Ebben az esetben természetesen merül fel az igény a jövőben rendelkezésünkre álló adat értékének ismeretére. Ezt írja le a tökéletes információ értéke (value of perfect informaiton, VPI), nagyon hasonlóan a 2 egyenlethez (lásd 3 ábra):

$$MEU(a^*|d) = \max_{a \in A} EU(a|d) = \max_{a \in A} \sum_{d_i \in \mathcal{D}} U(d_i) P(d_i|a,d), d \in \mathcal{D}, \tag{7}$$

ahol d a rendelkezésre álló adatot jelenti, $\mathcal{D}$ pedig minden lehetséges adathalmaz összessége, míg d_i az az adat, amihez akkor jutunk, ha az a lépést választjuk és ezzel pl. folytatjuk az adatgyűjtést. A 7 egyenlet valójában csak annyiban tér el a 2 egyenlettől, hogy az s-t a d helyettesíti, vagyis jelenleg a rendszer állapotát (a világról gyűjtött információt) a rendelkezésre álló adattal írjuk le. Fontos megjegyezni, hogy $d \subset d_i$. Ha rendelkezésre állna d_i információ, akkor a várható hasznosság a következőképpen alakulna:

$$MEU(a^*|d_i) = \max_{a \in A} EU(a|d_i) = \max_{a \in A} \sum_{d_j \in \mathcal{D}} U(d_j) P(d_j|a,d_i), d_i \in \mathcal{D}. \tag{8}$$

3. ábra. Az információ értéke

A 7 és a 8 kifejezések által definiált értékek eltérése adná meg a kívánt mennyiséget, vagyis annak a plusz információnak a hasznosságát, amit $d_i \setminus d$ ismerete jelent. Azonban d_i nem áll rendelkezésünkre, ezért jelölje D a jövőbeli adatot reprezentáló valószínűségi változót, így VPI nem más, mint a MEU jövőbeli várhatóértékének és a MEU különbsége:

$$VPI_d(D_j) = \left[\sum_i P(D_j = d_{ij}|d) MEU(a^*_{D_j,d}|D_j = d_{ij}, d)\right] - MEU(a^*_d|d), \qquad (9)$$

ahol a^*_d a d adat ismeretében legjobb döntés (lépés).

A VPI tulajdonságai

1. A VPI nem vehet fel negatív értéket,

$$VPI_d(D_i) \geq 0,$$

szemléletesen azért, mert az újonnan megszerzett információtól mindig el lehet tekinteni. Mivel a MEU érték egy maximum képzés eredménye, ezért amennyiben bármely újabb d_i információ mellett a MEU kisebb értéket venne fel, a maximum képzés miatt d_i-t üres adatnak kell feltételezni, hogy a legjobb eredményt kapjuk, így a VPI_d érték nullának adódik.

2. Könnyen belátható, hogy az információ értéke az adatok beérkezésének sorrendjétől független, és az információ értéke számolható a következőképpen:

$$VPI_d(D_i, D_j) = VPI_d(D_i) + VPI_{d,D_i}(D_j) = VPI_d(D_j) + VPI_{d,D_j}(D_i). \qquad (10)$$

3. Nem igaz azonban, hogy az információ értékének képzése additív

$$VPI_d(D_i, D_j) \neq VPI_d(D_i) + VPI_d(D_j),$$

hiszen pl. abban az esetben, ha a D_i és D_j valószínűségi változók azonos eloszlásúak $VPI_d(D_i, D_j) = VPI_d(D_i) = VPI_d(D_j)$.

Az információ értékének közelítése több megfigyelés esetén

Az információ értékével kapcsolatban eddig egy olyan esetet vizsgáltunk, amikor minden lépést megelőzően egyetlen valószínűségi változó jövőbeli várhatóértékét számítottuk ki. A gyakorlati esetek többségében a VPI számításához a 0.1.3 fejezetben tárgyalt egyváltozós módszert használják. Előfordulhat azonban olyan eset, amikor a döntési lépést megelőzően több valószínűségi változó értékét is meg kell becsülni. A 9 és a 10 egyenletekből könnyen levezethető, hogy a VPI kiszámításához szükséges idő több D_i valószínűségi változó esetében, azok n számával exponenciálisan arányos.

Ebben a fejezetben több megfigyelés együttes információértékét becsüljük az eddig ismertetettektől eltérő döntéselméleti modell mellett (4 ábra). Tegyük fel, hogy a $D_i, i = 1, \ldots, n$ valószínűségi változók függetlenek és a döntési lépést egy A bináris valószínűségi változóval modellezünk. Tegyük fel továbbá, hogy egy ismert eloszlású, bináris H valószínűségi változóval magasabb szinten tudjuk leírni a rendelkezésünkre álló $D_i, i = 1, \ldots, n$ adatot, vagyis ismertek a $P(D_i|H)$ feltételes eloszlások. Az A lépés és a H hipotézis közös hasznosságfüggvénye $U(A, H)$. Ezzel a modellel egy közelítő becslés adható a rendelkezésre álló változóhalmaz információértékére vonatkozóan lineáris időben.

Látni fogjuk, hogy a bizonyítás során nagyban kihasználjuk azokat az egyszerűsítéseket, amiket az A változó bináris volta és a szintén bináris H hipotézis változó jelent. Utóbbi felfogható, mint a rendelkezésre álló D_i adatok egy absztrakt, egyszerűsített leírása. Bár ez az egyszerűsített modell szükséges a többszörös megfigyelés információértékének lineáris időben történő kiszámításához, mégsem a valóságtól elrugaszkodott példa: képzeljük el, hogy a D_i változók egy beteg különböző leleteit reprezentálják, míg a H változó azt a feltételezést, hogy a beteg a leletek alapján egy súlyos betegségben szenved. Ha az A döntés a műtét elrendelését jelenti, akkor az $U(A, H)$ hasznosság jellemzi, hogy a betegség esetleges megléte mellett mennyire kockázatos, vagy hasznos a műtét, vagy annak elkerülése.

4. ábra. Információ értékének becslése több megfigyelés esetén

Ha felírjuk a H hipotézisre vonatkozó feltételes valószínűségek hányadosát (odds), akkor az a Bayes szabálynak ás a D_i valószínűségi változók függetlenségének köszönhetően átalakítható a következőképpen:

$$\begin{aligned} O(H|D_1, \ldots, D_n) &= \frac{P(H|D_1, \ldots, D_n)}{P(\neg H|D_1, \ldots, D_n)} \\ &= \frac{P(D_1|H)}{P(D_1|\neg H)} \cdots \frac{P(D_n|H)}{P(D_n|\neg H)} \frac{P(H)}{P(\neg H)} \\ &= O(H) \prod_{i=1}^{n} \lambda_i, \end{aligned} \tag{11}$$

ahol $\lambda_i = \frac{P(E_i|H)}{P(E_i|\neg H)}$ és $O(H) = \frac{P(H)}{P(\neg H)}$.

Legyen p^* a H hipotézis bekövetkezésének valószínűsége, amikor indifferens a döntéshozó számára mely lépést választja, formálisan:

$$p^* U(H, A) + (1 - p^*) U(\neg H, A) = p^* U(H, \neg A) + (1 - p^*) U(\neg H, \neg A). \tag{12}$$

A INFORMACIO ERTEKENEK KOZELITESE TOBB MEGFIGYELES ESETEN1 12 egyenletet átrendezve a p^* valószínűségre a következő érték adódik a hasznosságok ismeretében:

$$p^* = \frac{U(\neg H, \neg A) - U(\neg H, A)}{U(\neg H, \neg A) - U(\neg H, A) + U(H, A) - U(H, \neg A)}. \tag{13}$$

Mivel p^* valószínűség a döntési küszöb, ezért a döntéshozó akkor választja A-t $\neg A$-val szemben, ha

$$P(H|D_1, \ldots, D_n) > p^*, \tag{14}$$

amit átírva a következőt kapjuk:

$$O(H|D_1, \ldots, D_n) > \frac{p^*}{1 - p^*}. \tag{15}$$

A INFORMACIO ERTEKENEK KOZELITESE TOBB MEGFIGYELES ESETEN1 11 egyenlet alapján a INFORMACIO ERTEKENEK KOZELITESE TOBB MEGFIGYELES ESETEN1 15 kifejezés átírható

$$\prod_{i=1}^{n} \lambda_i > \frac{p^*}{1 - p^*} / O(H). \tag{16}$$

Ha a INFORMACIO ERTEKENEK KOZELITESE TOBB MEGFIGYELES ESETEN1 16 mindkét oldalának természetes alapú logaritmusát vesszük, akkor

$$\sum_{i=1}^{n} w_i > ln\frac{p^*}{1 - p^*} - lnO(H), \tag{17}$$

ahol $w_i = ln\lambda_i$, így definiálható a W valószínűségi változó, mint w_i változók összege

$$W \equiv \sum_{i=1}^{n} w_i, \tag{18}$$

és felírható a W változóhoz tartozó döntési küszöbérték:

$$W^* \equiv ln\frac{p^*}{1 - p^*} - lnO(H), \tag{19}$$

vagyis a döntéshozó akkor dönt A lépés mellett, ha

$$W > W^*. \tag{20}$$

W valószínűségi változó a w_i független valószínűségi változók összege, ezért a centrális határeloszlás tétele alapján eloszlása normális és várhatóértéke a w_i változók várhatóértékének összege, míg szórása a w_i változók szórásának összege, így:

$$p(W|H) \sim N(E(W|H), Var(W|H)). \tag{21}$$

A 21 egyenlet alapján kiszámítható, hogy mi annak valószínűsége, hogy a W valószínűségi változó a küszöbérték felett lesz:

$$p(W > W^*|H) = \frac{1}{\sigma\sqrt{2\pi}} \int_{W^*}^{\infty} e^{\frac{-(t-\mu)^2}{2\sigma}} dt. \tag{22}$$

0.2. Megállási feladatok

Ahogy azt az előző fejezetben láttuk, szekvenciális döntési helyzetben egy lehetséges lépés a leállás. Minden olyan esetben, amikor továbblépés valamilyen költséggel jár, minden lépésben optimális döntés lehet a megállás.

Egy szekvenciális kiválasztási probléma esetében a döntéshozónak egy n hosszú, szekvenciálisan érkező $X_1, \ldots, X_n$ változó sorozatból ki kell választani a legnagyobbat úgy, hogy a be nem érkezett változókról semmilyen információja nincs, korlátozottan választhat a már beérkezettek közül, és a játék bizonyos variációiban n végtelen, vagy nincs róla információ. Az egyik legegyszerűbb megállási probléma a Titkárnő probléma.

0.2.1. Titkárnő probléma

A megállási problémák alapfeladata a titkárnő probléma, egy munkáltatónak a legmegfelelőbb munkaerőt kell kiválasztania egy pozícióra. A feladat a következő szabályokkal definiálható:

1. Csak egyetlen szabad állás van.
2. A jelentkezők száma, n, előre ismert.
3. Az interjúk egymás után, egyesével történnek.
4. A jelentkezők meghallgatása véletlenszerű sorrendben történik, minden sorrend egyformán valószínű.
5. Minden interjú után az addig meghallgatott jelentkezők alkalmasságuk szerint egyértelműen rendezhetők.
6. Minden interjú után el kell dönteni, hogy a jelentkezőt felveszik-e, vagy sem. Ha egy jelentkezőt nem vesznek fel, nem lehet többé visszahívni.
7. A munkáltatónak csak a legalkalmasabb jelölt felel meg, minden más jelölt azonos mértékben alkalmatlan.

Vagyis a döntéshozó igen nehéz helyzetben van, mert a már elküldött jelentkezőkről bár van információja nem tudja visszahívni, azokról a jelentkezőkről pedig, akik még nem voltak interjún semmilyen információja nincsen. A probléma megoldása ebből a felismerésből adódik: a döntéshozó minden esetben csak a már meglévő információ alapján dönthet, vagyis érdemes megfelelő információt begyűjteni, hogy aztán ezek alapján az információk alapján a lehető legnagyobb valószínűséggel el lehessen dönteni egy jelentkezőről, hogy az a legjobb-e. A megoldásként adódó algoritmus:

5. ábra. Titkárnő probléma

1. Az első $r-1$ jelentkező meghallgatása után,

2. azt a jelentkezőt kell választani, amelyik jobb, mint az első $r-1$ jelentkező bármelyike.

Annak valószínűsége, hogy adott r mellett a fenti algoritmussal a legjobb jelentkezőt választjuk:

$$P_{opt}(r) = P(r \text{ mellett a legjobbat választjuk}) = \sum_{i=r}^{n} \frac{1}{n} \frac{r-1}{i-1}, \tag{23}$$

mivel az $\frac{r-1}{i-1}$ hányados annak a feltételes valószínűségét adja, hogy ha i a legjobb jelölt, akkor az előző $i-1$ jelentkező közül a legjobb az első $r-1$ jelentkező között van. Minden esetben a 23 kifejezést maximalizáló r-t kell választani.

Bizonyítható, hogy n növekedtével az optimális r tart n/e -hez, és annak a valószínűsége, hogy az algoritmus a legjobb jelentkezőt választja tart $1/e$-hez, ahol e az Euler szám. Vagyis annak a valószínűsége, hogy megtaláljuk a legjobb jelöltet megközelítőleg 0.368.

Az állítás bizonyításához először átalakítjuk a 23 kifejezést, majd belátjuk, hogy $n \to \infty$ esetén $P_{opt}(r) \to -xln(x)$, aminek szélsőértéke könnyen meghatározható. Első lépésben a $P_{opt}(r)$ átalakítása:

$$\begin{aligned} P_{opt}(r) &= \sum_{i=r}^{n} \frac{1}{n} \frac{r-1}{i-1} && (24) \\ &= \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1} \\ &= \frac{r-1}{n} \sum_{i=r}^{n} \frac{n}{i-1} \frac{1}{n}. && (25) \end{aligned}$$

Egy tetszőleges $f(\cdot)$ függvény bal oldali Riemann összege az $[a,b]$ intervallumon:

$$\Delta x(f(a) + f(a+\Delta x) + f(a+2\Delta x) + \ldots + f(b-\Delta x)) = \sum_{i=0}^{\frac{b-a}{\Delta x}} f(a+i\Delta x)\Delta x. \tag{26}$$

A fenti egyenletbe helyettesítve $f(t) = 1/t$ függvény esetén $\Delta x = 1/n$ lépésközzel, ahol $a = \frac{r-1}{n}$ és $b = 1$, nem más, mint

$$\sum_{i=0}^{\frac{b-a}{\Delta x}} f(a+i\Delta x)\Delta x = \sum_{i=0}^{n-r+1} \frac{n}{(r-1+i)}\frac{1}{n} = \sum_{i=r-1}^{n} \frac{n}{i-1}\frac{1}{n}. \quad (27)$$

Tehát a 27 $n \to \infty$ esetén

$$\lim_{n\to\infty} \sum_{i=r-1}^{n} \frac{n}{i-1}\frac{1}{n} = \int_{\frac{r-1}{n}}^{1} \frac{1}{t}dt. \quad (28)$$

A 28 határértéket visszaírva a 24 egyenletbe az $x = \frac{r-1}{n}$ helyettesítéssel

$$\lim_{n\to\infty} \frac{r-1}{n} \sum_{i=1}^{n} \frac{n}{i-1}\frac{1}{n} = x\int_{x}^{1} \frac{1}{t}dt = -xln(x). \quad (29)$$

Mivel $-xln(x)\frac{dx}{dt} = 1 + ln(x)$, ami az $x = 1/e$ helyen veszi fel a 0 értéket, ezért bizonyítottuk az eredeti állítást.

0.2.2. A Googol játék

A Googol játék a megállási problémák egyik első verziója, amit 1960-ban Martin Gardner publikált. A Googol játékban ketten vesznek részt, az egyik szereplő előre meghatározott n számú lapra felír általa választott, különböző egész számokat, a másik szereplő az eddigi döntéshozó helyzetében van: a lefordított lapok közül addig húz, amíg úgy nem gondolja, hogy a legnagyobb számot tartalmazó lapot tartja a kezében. Ez a feladatkiírás annyiban tér el a korábbitól a döntéshozó szemszögéből, hogy nem feltételezheti az egymás után érkező elemek függetlenségét. A számokat meghatározó személy ellenségesen is viselkedhet.

Ha elfogadjuk a feltételezést, hogy az előre kiválasztott n szám együttes eloszlása leírható egy egyváltozós sűrűségfüggvénnyel, melynek egyetlen argumentuma az n szám maximuma

$$p(x_1, \ldots, x_n) = g(max\{x_1, \ldots, x_2\}), \quad (30)$$

vagyis ha az egyenlet által meghatározott értelemben az $\{X_1, \ldots, X_n\}$ számsorozat felcserélhető, akkor bizonyítható, hogy $n > 2$ esetén a Googol játék megoldása hasonló algoritmus alapján történik, mint a Titkárnő problémáé, ahol r a következőképpen adódik:

$$\frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{n-1} < 1 < \frac{1}{r-1} + \frac{1}{r} + \ldots + \frac{1}{n-1}. \quad (31)$$

Az is belátható, hogy a Googol játékot játszó személy számára a következő két eset ekvivalens:

1. Nem tud semmit a lapokon található számokról (akár determinisztikusak is lehetnek)

2. A számok egyenletes eloszlásúak a $(0, \beta)$ intervallumon, ahol β ismeretlen.

0.2.3. Odds algoritmus

Az Odds algoritmus több megállási probléma megoldását adja azzal, hogy egy általánosabban megfogalmazott feladatot old meg: adott $I_1, \ldots, I_n$ indikátorváltozó sorozat, ahol I_j változó A_j esemény bekövetkezését mutatja. Az események egymás után, egyesével következnek be, a cél egy olyan módszert megadni, ami biztosítja, hogy a döntéshozó a legnagyobb valószínűséggel álljon meg az utolsó bekövetkező eseménynél. A $max_t\{P(I_t = 1, I_{t-1} = 0, \ldots, I_1 = 0)\}$ kifejezést maximalizáló index a τ leállási idő.

Belátható, hogy ha I_j bekövetkezésének valószínűsége p_j és $o_j = p_j/(1 - p_j)$ (odds, arány), akkor τ az első olyan index, ahol $I_\tau = 1$, $\tau > r_n$ és

$$r_n = \max\{1, \max\{1 \leq k \leq n : \sum_{j=k}^{n} o_i \geq 1\}\}. \tag{32}$$

Vagyis az r index, pontosan akkor optimális, ahonnan a hátralévő odds-ok összege először nagyobb, mint 1, vagy ha nincs ilyen index, akkor az $r = 1$ érték. Annak valószínűsége, hogy az eljárással az első „sikeres" eseménynél állunk meg:

$$P(I_\tau = 1, I_{t-1} = 0, \ldots, I_1 = 0) = \left(\prod_{j=r}^{n}(1 - p_j)\right)\left(\sum_{j=r}^{n} o_j\right). \tag{33}$$

Az Odds algoritmussal megoldható a titkárnő probléma, ha az eredeti feladatot átfogalmazzuk a következőképpen: $I_k = 1$, ha a k sorszámú jelentkező jobb, mint a korábbiak $(X_k > X_i, \forall i < k)$, így $P(I_k) = 1/k$, $o_k = 1/(k - 1)$. Tehát az optimális r, ahol a $R = 1/(n-1) + 1/(n-2) + \ldots + 1/(r-1)$ összeg nagyobb lesz, mint 1. Ha $n \to \infty$, akkor $R \to 1/e$, ahogy korábban is láttuk.

0.2.4. Az Odds algoritmus egy folytonos kiterjesztése

Időben többször bekövetkező független események között eltelt idő modellezésére használt valószínűségi változó eloszlása folytonos esetben exponenciális eloszlású, mivel a folytonos eloszlások közül ez az egyetlen örökifjú tulajdonságú

$$P(X_T > x_s + x_t | X_T > x_s) = P(X_T > x_t), \forall x_s, x_t > 0. \tag{34}$$

A 34 egyenlet szemléletesen annyit jelent, hogy ha x_s ideje állunk a buszmegállóban, annak valószínűsége, hogy a további x_t időintervallumban sem fog busz érkezni, nem függ x_s-től, vagyis nem függ attól mennyi ideje várunk. Ha a buszok érkezését független eseményként kezeljük, akkor az előző busz érkezése semmilyen hatással nincs a következő busz érkezésére.

Bár a buszok érkezése a menetrend ellenére jó közelítéssel független, további példa lehet ilyen folyamatokra egy kevésbé forgalmas úton közlekedő autók közti távolság, vagy a beérkező telefonhívások közt eltelt idő.

Ha valószínűségi változók közt eltel idő λ paraméterű exponenciális eloszlású,

$$f(x;\lambda) = \left\{ \begin{array}{ll} \lambda e^{\lambda x} & \text{ha } x > 0, \\ 0 & \text{egyébként,} \end{array} \right\} \tag{35}$$

akkor annak valószínűsége, hogy τ időintervallumban az adott esemény k alkalommal fordul elő, λ paraméterű homogén Poisson eloszlást követ:

$$P[(N(t+\tau) - N(t)) = k] = \frac{e^{-\lambda\tau}(\lambda\tau)^k}{k!} \qquad k = 0, 1, \ldots, \tag{36}$$

ahol $N(t)$ a t időpillanatig bekövetkezett események száma. Ha a λ intenzitás paraméter időben változhat, azaz $\lambda(t)$, akkor inhomogén Poisson folyamatról beszélünk.

Ha a $\lambda(t)$ paraméterű Poisson eloszlású valószínűségi változók „sikerességét" a $h(t)$ sűrűségfüggvény írja le, akkor a megállási probléma a következőképpen módosul: állítsuk meg a játékot a $[0, T]$ időintervallumban az utolsó sikeres eseménynél. A folytonos feladat a következőképpen oldható meg: a $[0, T]$ időintervallumot m részre osztva, annak valószínűsége, hogy a k sorszámú intervallumban legalább egy sikeres esemény következik be $p_k = \lambda(t_k)h(t_k)(t_k - t_{k-1}) + o(t_k - t_{k-1})$. Ha az intervallumok számát növelve, azok mérete tart a nullához, $(t_k - t_{k-1}) \to \infty$, akkor az intervallumban a bekövetkezés valószínűsége tart az intenzitás és a sikeresség valószínűségének szorzatához, $p_k \to \lambda(t_k)h(t_k)$. Ezek alapján a diszkrét esethez nagyon hasonló a megoldás:

$$\tau = \sup\left\{0, \sup\left\{0 \le t \le T : \int_t^T \lambda(u)h(u)du \ge 1\right\}\right\} \tag{37}$$

0.3. Többkarú rabló feladatok

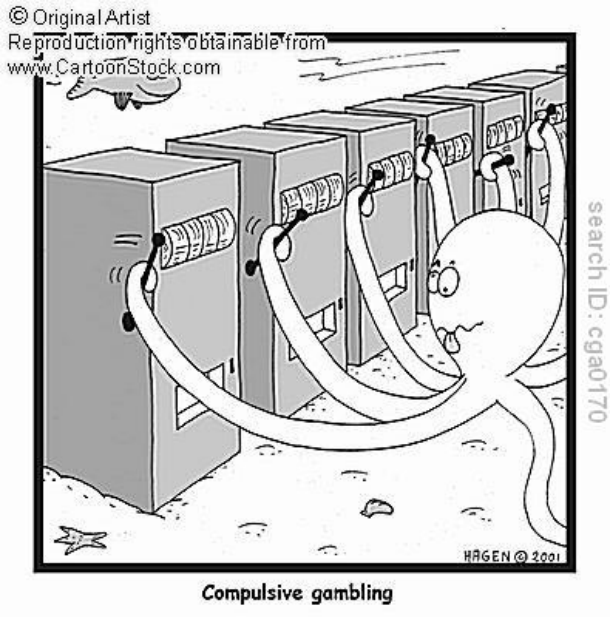
A többkarú rabló probléma (multi-armed bandit problem, MAB) egy erőforrás allokációs probléma. Alapfeladata megfeleltethető egy szerencsejátékos problémájának: a játékos k félkarú rabló előtt áll, és szeretné maximalizálni a várható nyereményét. A játékos minden lépésben választ egy játékautomatát, melynek meghúzza a karját. Az a gép, amelynek meghúzzuk a karját, egy, a gépre és annak pillanatnyi állapotára jellemző valószínűségi eloszlás szerint fizet jutalmat. A valós helyzettől a többkarú rabló alapprobléma annyiban tér el, hogy nincs költsége a gépek működtetésének. A cél minden esetben az erőforrások optimális kihasználása: véges horizonton a begyűjtött jutalmak összegének maximalizálása, végtelen horizonton adott diszkontrátával, vagy végtelen horizonton átlagban.

A többkarú rabló probléma k független karból/folyamatból/gépből és egy kontroller folyamatból áll (a három fogalmat: kar, folyamat, gép a fejezetben mostantól felváltva használjuk). Minden karhoz két véletlen folyamat tartozik $(X(0), X(1), \ldots,, R(X(0)), R(X(1)), \ldots,)$, ahol az $X(n)$ a kar állapota azután, hogy a kart n-szer működtettük, $R(X(n))$ pedig az $X(N)$ állapotért kapott jutalom. Egy gép állapota a következőképpen változik:

$$X(n) = f_{n-1}(X(0), \ldots, X(n-1), W(n-1)),$$

ahol $f(\cdot)$ adott és $W(n)$ egy ismert eloszlású, valós-értékű, független, azonos eloszlású valószínűségi változó sorozat, mely független $X(0)$-tól. Mivel az $f(\cdot)$ függvény determinisztikus, a $W(n)$ valószínűségi változó jelenti a véletlen faktort a gép működésében.

A k-karú rabló probléma k karja független egymástól, a karokat egy kontroller/processzor folyamat működteti: minden diszkrét időpillanatban egyet és csak egyet választva ki. A kiválasztott folyamat állapotot vált, a többi állapota változatlan marad. A cél a várható jutalom maximalizálása.



6. ábra. Ábra címe ????

0.3.1. Alkalmazási területek

1. Szenzor menedzsment – egy egyszerűsített példa azt szemlélteti, hogy egy szenzorral hogyan keresünk egy célpontot, mely k lehetséges dobozok egyikében van. A szenzorunk képes érzékelni a célpontot, de csak bizonyos bizonytalansággal. Célunk egy előre meghatározott küszöbnél nagyobb bizonyosság elérése, vagyis minden lépésben a jutalom az a jel (pl. 0 és 1 közötti), amit a szenzor kibocsát. Azzal tehát, hogy a várható jutalmat növeljük, a célpontot keressük.

2. Online hirdetések kiválasztása – a feladat ebben az esetben az, hogy a hirdetést közvetítő (advertisor) kiválassza a hirdetők (auctioneer) közül azt, akinek a hirdetésére a lehető legtöbben kattintanak. Bár a hirdetőnek vannak adatai arról, hogy máshol mennyien kattintottak a hirdetésére, lehetséges, hogy a valós számnál nagyobbat közöl a hirdetést közvetítő felé. A megfelelő hirdető kiválasztása a hirdetés

közvetítő számára egy többkarú rabló probléma. Hasonlóan, mint az előző példánál, ha a jutalom a kattintások száma, akkor a várható jutalom maximalizálásával a „legjobb" reklámokat választjuk ki.

3. Online keresés – online tartalmak keresése során megfigyelhető, hogy bizonyos keresési kifejezésekre elvárt találati eredmények tematikája idővel jelentősen módosulhat: jó példa erre a napi hírek területe. Ha pl. napokkal egy atomkatasztrófa után keresek a „nuclear accident" kulcsszóra, joggal várom el, hogy friss hírekhez és információkhoz jussak, míg mondjuk korábban az előző atomkatasztrófák találatait várjuk el. Ezeket a találati „eltolódásokat" a hagyományos keresési algoritmusokra alapozott többkarú rabló meta-algoritmussal lehet feltérképezni.

4. Sorbanállási és ütemezési feladatok – a MAB feladat felfogható egy egyetlen processzorból, és k feladatból álló rendszernek, ahol minden egyes lépésben el kell dönteni, hogy a processzor mely feladatot hajtsa végre. Minden feladat végrehajtásáért jár egy jutalom, ami pl. a feladat sürgősségét tükrözi.

5. Klinikai kísérlettervezés – a gyógyszerkísérletek esetén a gyógyszerek hatóanyagainak megválasztása szintén megfeleltethető a MAB problémának. Itt az egyes karokat a hatóanyagok jelentik, míg a jutalmat a betegek állapota, esetleg túlélési rátája: a legnagyobb várható jutalomtól a legjobb hatóanyag kiválasztását reméljük.

0.3.2. Az optimális megoldás, előrefele következtetés

A MAB probléma esetében a visszafele következtetés (backward induction) minden esetben optimális megoldást ad, viszont rendkívül számításigényes, ezért a gyakorlatban igen ritkán alkalmazzák.

Az előrefele következtetés legegyszerűbb formája az úgynevezett rövidlátó (myopic) előretekintés, egyetlen lépésre előre maximalizálja a jutalmat. Ez a megoldás általában nem vezet optimális megoldáshoz. Az előrefele következtetés bonyolultabb típusa T lépésre előre számítja a várható jutalmat és ezt az értéket próbálja maximalizálni. Utóbbi megoldással az esetek nagy részében csupán szuboptimális megoldáshoz jutunk.

A T lépéses előretekintés kiterjesztése, amely esetében feltételezzük, hogy egy végrehajtási stratégia adott, és az ismert végrehajtási stratégia függvényében határoz meg egy τ „leállási" időt, melyre a végrehajtási folyamat a maximális várható jutalmat adja. A végrehajtást csak a meghatározott leállási időpillanatig folytatjuk, az optimalizációt csupán erre kell végrehajtani.

Az előrefele következtetés a τ leállási idő számításával a következőképpen alakul:

1. Ki kell választani egy stratégiát, a stratégia ebben az esetben egyetlen gép működtetését jelenti.

2. Ki kell számítani egy τ leállási időt

3. A τ leállási időig követjük az 1. pontban választott stratégiát. τ után az 1. lépéssel folytatjuk.

Általános esetben a fent leírt stratégia sem vezet optimális megoldáshoz, azonban az alábbi feltételek mellett az algoritmus optimális a MAB problémára:

1. A kontroller folyamat egy időben csak egyetlen gépet üzemeltet; az üzemeltetett gép állapota nem befolyásolható, csak ki- és bekapcsolni lehet.

2. A nem működtetett gép nem vált állapotot.

3. A gépek függetlenek

4. A nem működtetett gépek nem adnak jutalmat.

A fent ismertetett algoritmus optimalitását, a megadott feltételek mellett szemléletesen úgy láthatjuk be, hogy minden alkalommal, amikor kiválasztunk egy gépet, majd azt τ-ig működtetjük, nem hozunk „visszafordíthatatlan" döntést. A többi gép állapota nem változik, vagyis nincs olyan jutalom, melyet hosszútávon az algoritmus ne tudna megszerezni, így az előrefele következtetés optimális megoldáshoz vezet.

0.3.3. Gittins index

Mivel nem biztos, hogy t időpillanatig egy folyamat t-szer vált állapotot (hiszen üzemeltethetjük a többi folyamatot is), ezért i folyamat állapotát t-ben $N_i(t)$-vel jelöljük. Egy folyamatot az állapot és a jutalom sorozata írja le: $(X_i(N_i(t)), R_i(N_i(t)))$; $N_i(t) = 1, 2, \ldots, t; t = 1, 2, \ldots$ és $i = 1, 2, \ldots, k$. $U(t)$ vektor jelöli, hogy t időpillanatban melyik folyamatot üzemelteti a kontroll folyamat. $U(t) = (U_1(t), \ldots, U_k(t)), U(t)$ minden időpillanatban egyetlen komponensében nem nulla, a komponens indexe az adott pillanatban üzemeltetett folyamat indexét jelöli.

A MAB alapfeladata, hogy maximalizálja a következő kifejezést:

$$J = E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{k} R_i \left[X_i(N_i(t), U_i(t)) | X_1(N_1(0)), \ldots, X_k(N_k(0))\right]\right], \tag{38}$$

ahol $0 < \beta < 1$, vagyis a jutalom várható jelenértékét maximalizáljuk.

A Gittins index a következő kifejezést takarja:

$$v_{x_i}(x_i(0)) = max_{0<\tau} \frac{E\left[\sum_{t=0}^{\tau-1} \beta^\tau R_i(X_i(t)|x_i(0))\right]}{E\left[\sum_{t=0}^{\tau-1} \beta^\tau |x_i(0))\right]}, \tag{39}$$

vagyis a Gittins index nem jelent mást, mint hogy minden egyes karra meghatározunk egy olyan τ megállási időt, amire nézve a fenti hányados maximális. Így a már leírt előrefele következtetés algoritmusa a következőképpen alakul:

1. Minden folyamatra kiszámítjuk a Gittins indexet, ezzel együtt minden folyamatra meghatározunk egy τ leállási időt.

2. Kiválasztjuk a maximális indexszel rendelkező folyamatot, majd az indexhez tartozó leállási ideig működtetjük. Leálláskor az első ponttal folytatjuk.

A fenti algoritmus Markovi feltételezés mellett optimális megoldáshoz vezet.