Luento 7: Oppiminen moniagenttijärjestelmissä

Raul Hakli

Oppivat agentit

- Älykkäiltä agenteilta vaaditaan usein *oppimiskykyä*. Tällä tarkoitetaan että agentin pitäisi pystyä muokkaamaan toimintaansa siten että 1) parantaa suoritustasoaan staattisessa ympäristössä tai 2) ylläpitää suoritustasoaan dynaamisessa ympäristössä.
- Monen agentin ympäristöt eivät ole vain määrällisesti monimutkaisempia vaan laadullisesti erilaisia ympäristöjä
- Tavanomaisessa dynaamisessa ympäristössä agentti voi tarkkaila tekojensa seurauksia tilastollisesti: useimmiten jos teen teon A, seuraa tapahtuma B. Monet teot muuttavat ympäristöä joten agentti voi päätyä seuraavanlaiseen sääntöön: yleensä tekoa A seuraa tapahtuma B noin 5 kertaa peräkkäin, mutta ei sitten enää.
- Monen agentin ympäristö ei vain muutu agentin toiminnan seurauksena vaan se pyrkii aktiivisesti mukautumaan siihen, ennustamaan sitä ja vaikuttamaan siihen. (Ensimmäisellä kerralla A:ta seurasi B, toisella C ja sen jälkeen A:ta ei enää voinut tehdä.)

Oppiminen ja opettaminen

- Oman käyttäytymisen muuttaminen johtaa siihen että ympäristökin muuttaa käyttäytymistään → yksinkertaisetkin oppimissäännöt tekevät järjestelmästä kompleksisen dynaamisen systeemin jonka käyttäytymistä on vaikea ennustaa.
- Tätä matemaattista tosiasiaa tärkeämpi on kuitenkin käsitteellinen seikka: moniagenttijärjestelmissä oppimista ei voi erottaa opettamisesta: teon valinnassa ei voi ottaa huomioon vain sitä mitä on opittu muiden agenttien aiemmasta käyttäytymisestä vaan myös se miten niiden tulevaan käyttäytymiseen halutaan vaikuttaa.
- Muut agentit on otettava huomioon intentionaalisina toimijoina, ei pelkkänä stokastisena prosessina.

Esimerkki

	L	R
T	1,0	3,2
B	2,1	4,0

Ykköspelaajalla (rivi) on dominanttistrategia B ja (B, L) on yksikäsitteinen Nash-tasapaino. Toistetussa pelissä ykköspelaaja voi kuitenkin ottaa opettajan roolin ja ryhtyä pelaamaan jatkuvasti T:tä, jolloin kakkospelaajan on järkevää vaihtaa taktiikkaa ja ryhtyä pelaamaan R:ää.

Esimerkki 2

	L	R
T	1,1	0,0
B	0,0	1,1

Tässä koordinaatiopelissä kumpi tahansa voi yrittää ottaa opettajan roolin, mutta jos molemmat tekevät sen ja sattuvat vielä valitsemaan niin että päätyvät nollatulokseen, niin sama voi jatkua loputtomiin. (Jos molemmat noudattavat samaa sääntöä ja yrittävät vaihtaa taktiikkaa samoin väliajoin, niin edelleen pysytään nollassa.) Onko olemassa oppimissääntöä, jolla agentit voisivat koordinoida toimintansa ilman ulkopuolista opettajaa?

Mitä oppiminen on?

- Toimintasäännön / käyttäytymisen muuttaminen kokemuksen perusteella?
- Esimerkki peliteoriasta: *toistetut pelit* (repeated games). Esimerkiksi tit-for-tat-strategia iteroidussa vangin dilemmassa
- stokastiset pelit yleensäkin
- suuret populaatiot ja samantapaisten tilanteiden toistuminen (vrt. iteroituihin peleihin)
- Oppimisen kohteena voi olla joko vain strategia tai koko pelitilanne (toimintavaihtoehdot, payoff-funktiot jne)

Miksi oppimista?

- deskriptiiviset oppimisteoriat: miten oppimista tapahtuu todellisessa elämässä (ihmiset, eläimet, organisaatiot)

  Oppimisteoriaa verrataan havaittavaan oppimiseen. Teoria on hyvä jos se selittää ja ennustaa todellista oppimista.
- normatiiviset oppimisteoriat: kuinka älykkäiden agenttien pitäisi oppia

  Mitkä ovat kriteerit hyvälle normatiiviselle oppimisteorialle?

Kriteereistä (1)

- Oppimisen tulos on optimaalinen? Tämä kriteeri on mielekäs vain rajoitetuissa tapauksissa kuten kahden agentin nollasummapeleissä. Yleisessä tapauksessa on vaikea sanoa mitä se tarkoittaa koska tulos ei riipu vain agentin omista toiminnoista vaan myös muiden (vrt. tasapainotilat peliteoriassa)
- Hannan-konsistenssi: jos oppimisprosessi konvergoi johonkin strategiaan, sen tulisi olla paras vastaus muiden agenttien strategiaprofiiliin.
- konvergenssi
- oppimisen nopeus

Kriteereistä (2)

Opituissa strategioissa on myös eroja sen suhteen, kuinka hyvin ne toimivat erilaisia agentteja vastaan.

- self-play: muut pelaajat käyttävät samaa strategiaa kuin itse
- strategian menestyminen riippuu muiden pelaajien oppimistavasta (muista oppiminen vs. opettaminen): esim. chicken

	luovuta	haasta
luovuta	2,2	1,3
haasta	3,1	0,0

Jos toiset ovat mukautuvia ja pyrkivät etsimään parasta vastausta, niin kannattaa omaksua opettajan rooli ja haastaa välittämättä siitä mitä muut tekevät. Jos toinenkin noudattaa tätä politiikkaa, niin tulos on pahin mahdollinen.

Esitetyt teoriat

- Moniagenttioppimisessa ei oikeastaan ole puhtaasti keskitytty joko deskriptiivisiin tai normatiivisiin teorioihin vaan yritetty jotenkin yhdistää niitä. Psykologisista oppimisteorioista on haettu malleja mutta on kuitenkin pyritty saavuttamaan tasapainotiloja.
- Ideaalina olisi kai teoria joka olisi sekä psykologisesti uskottava että todistettavasti Nash-tasapainoon konvergoituva.
- Tulokset ovat tähän mennessä jääneet puutteellisiksi: oppimismekanismit ovat kovin yksinkertaisia verrattuna ihmisten oppimiseen ja silti usein saadaan vain osittaisia konvergenssituloksia.

Palauteoppiminen

- Tarkastellaan yhden agentin *palauteoppimista* tai *vahvistusoppimista* (reinforcement learning)
- Mainitaan usein kolmantena koneoppimisen perustyyppinä ohjatun (supervised) ja ohjaamattoman (unsupervised) ohella.
- Perusidea: agentti kerää tietoa tekojensa seurauksista ja pyrkii valitsemaan useammin sellaisia tekoja joiden seuraukset ovat aiemman kokemuksen mukaan olleet hyviä.
- Tasapainottelu hyväksi havaittujen tekojen hyödyntämisen (exploitation) ja uusien valintojen kartoittamisen tai tutkimisen (exploration) välillä

Oppiminen ja MDP:t (1)

Palautetaan MDP:t mieliin toiselta luennolta:

- agentti tekee peräkkäisiä päätöksiä $a_t, a_{t+1}, \ldots$ kullakin ajanhetkellä t havaittavassa, stokastisessa ympäristössä
- kussakin tilassa agentti saa *palkkion* (reward) $R(s_t) \in \mathbb{R}$ ja kutakin toimintoa seuraa tilan s_t vaihtuminen uudeksi tilaksi s_{t+1}
- agentin tehtävänä on maksimoida tulevien palkkioiden diskontattua summaa $R(s_t) + \gamma R(s_{t+1}) + \gamma^2 R(s_{t+2}) + \ldots$, missä $\gamma \in [0, 1]$ on *diskonttotekijä* (discount factor)
- palkkion suuruus riippuu agentin noudattamasta *politiikasta* (policy) π, joka määrää minkä toiminnon agentti kussakin tilassa valitsee
- optimaalinen politiikka π^* on sellainen joka maksimoi tulevien palkkioiden odotusarvoa

Oppiminen ja MDP:t (2)

- Määritellään tiloille arvofunktio $V^\pi : S \to \mathbb{R}$, joka kertoo tilan arvon kun käytetään politiikkaa π:

$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s\right]$$

- Vastaavasti määritellään tila-toiminto-pareille arvofunktio $Q^\pi : S \times A \to \mathbb{R}$, käytettäessä politiikkaa π:

$$Q^\pi(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_t = s, a_t = a\right]$$

- Näille saadaan seuraava yhteys:

$$Q^\pi(s, a) = R(s) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = Q^\pi(s, \pi(s)).$$

Oppiminen ja MDP:t (3)

- Optimaalisella politiikalla π^* arvofunktiot ovat seuraavat:

$$V^*(s) = \max_\pi V^\pi(s) \qquad Q^*(s, a) = \max_\pi Q^\pi(s, a)$$

kaikilla $s \in S$ ja $a \in A(s)$.

- Optimaalisen politiikan *Bellmanin yhtälöt* V:lle ja Q:lle suhteuttavat tilojen arvot niiden seuraajatilojen arvoihin:

$$V^*(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) V^*(s').$$

$$Q^*(s, t) = R(s) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

Arvojen iterointi

Jos siirtymämalli $P(s'|a, s)$ on agentin tiedossa, tilojen optimaaliset utiliteetit voidaan laskea käyttämällä *arvojen iterointia*:

1. Aluksi asetetaan kaikille tiloille s jotkin (esim. satunnaiset) alkuarvot $V(s)$.
2. Toistuvasti päivitetään tilojen utiliteettiarvoja soveltamalla Bellmanin yhtälöön perustuvaa päivityssääntöä:

$$V(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) V(s').$$

3. Lopetetaan päivittäminen kun arvot *konvergoituvat* eli suppenevat kohti jotakin arvoa niin että muutokset ovat enää hyvin pieniä.

Tilojen utiliteetit

Arvojen iterointi johtaa tilojen arvojen konvergoitumiseen kohti optimaalista utiliteettifunktiota $V^*(s)$.

Toisen luennon esimerkkiympäristössä tilojen utiliteetit ovat seuraavat:

3	0.812	0.868	0.918	+1
2	0.762		0.660	−1
1	0.705	0.655	0.611	0.388
	1	2	3	4

(AIMA, Fig. 17.3)

Optimaalinen politiikka pyrkii maksimoimaan utiliteettia ja tässä siis yrittää siirtyä siihen naapuriruutuun, jonka utiliteetti on suurin.

Oppiminen palautteesta

- Arvojen iterointi perustuu dynaamiseen ohjelmointiin ja sitä voidaan käyttää kun MDP:n siirtymämalli $P(s'|a,s)$ ja palkkiofunktio $R(s)$ tunnetaan.
- Tavallisesti agentit ovat kuitenkin vieraassa ympäristössä ja joutuvat oppimaan kokemuksistaan vuorovaikutuksessa ympäristön kanssa $\rightarrow$ palauteoppiminen.
- Osoittautuu, että optimaalinen politiikka löydetään myös tuntematta siirtymätodennäköisyyksiä kunhan ympäristön käyttäytymisestä kerätään riittävästi "näytteitä" kokeilemalla minkälaisia seurauksia eri teoilla on eri tiloissa.

Q-oppiminen (1)

Erityisen käyttökelpoinen palauteoppimisen muoto on *Q-oppiminen*, jossa päivitetään *Q-funktiota*, joka arvioi tekojen utiliteettia eri tiloissa.

1. Agentti aloittaa satunnaisilla alkuarvoilla $Q(s,a)$ kullekin tila-toiminto-parille.
2. Agentti tutkii ympäristöä kokeilemalla nykyisessä tilassa s jotakin toimintavaihtoehtoa a, päätyy uuteen tilaan s' ja saa palkkion R.
3. Agentti päivittää Q-funktion arvoa seuraavasti:

$$Q(s,a) \leftarrow (1-\lambda)Q(s,a) + \lambda[R + \gamma \max_{a'} Q(s',a')],$$

   missä parametri $\lambda \in [0,1]$ säätelee konvergoitumisen nopeutta ja a' käy läpi uuden tilan s' toimintavaihtoehdot.
4. Palataan kohtaan 2 kunnes Q-arvot konvergoituvat.

Q-oppiminen (2)

- Oppimisvaiheessa tekojen valinta on aluksi satunnaista tutkiskelua, mutta vähitellen opittuja asioita voidaan hyödyntää pyrkimällä suosimaan niitä tekoja, joiden Q-arvo on suuri.
- Voidaan osoittaa, että jos kaikkia tila-toiminta-pareja kokeillaan äärettömän monta kertaa ja λ pienenee hitaasti kohti nollaa, Q-arvot konvergoituvat kohti optimaalista Q-funktiota $Q^*(s,a)$.
- Optimaalinen politiikka valitsee sitten yksinkertaisesti kussakin tilassa ahneesti sen toimintavaihtoehdon, jonka Q-arvo on suurin.
- Q-oppiminen takaa hyvän tuloksen, mutta oppiminen voi olla hyvin hidasta, diskontatuilla palkkioilla liian hidasta. Ongelma voidaan välttää, jos opetusvaihe voidaan suorittaa simuloidussa ympäristössä ennen kuin agentti päästetään todelliseen ympäristöön.

Monen agentin palauteoppiminen

- Ei ole aivan selvää, kuinka palauteoppimista pitäisi soveltaa monen agentin tapauksessa. Monenlaista on kuitenkin kokeiltu.
- Jotkut menetelmät on tarkoitettu kooperatiiviseen tapaukseen, jossa agentit pyrkivät yhdessä oppimaan kaikkia hyödyttävän yhteisen toimintaprofiilin. Toiset taas keskittyvät yhden agentin oppimiseen kilpailutilanteessa.

Kooperatiivinen oppiminen (1)

- Kooperatiivisessa oppimisessa agenteilla on yhteinen palkkiofunktio ja ne pyrkivät oppimaan palkkiota maksimoivan politiikan joka kertoo kaikkien agenttien toiminnat kussakin tilanteessa, siis politiikan $\pi^*(s) = (\pi_i^*(s))$.
- Yksinkertaisin tapa on että agentit oppivat toisistaan *riippumattomasti* jolloin ne pitävät muita agentteja osana ympräistöä eivätkä yritä mallintaa tai ennustaa niiden toimintaa.
- Tällöin voidaan käyttää tavallista Q-oppimista vaikka sitä ei tällaiseen tilanteeseen olekaan tarkoitettu. Konvergenssia ei enää voi taata, mutta käytännössä on todettu että oppiminen usein konvergoituu ja joskus jopa Nash-tasapainoon.

Kooperatiivinen oppiminen (1)

- Toinen mahdollisuus on, että agentit ottavat huomioon kaikkien muidenkin agenttien toiminnot ja päivittävät funktiota $Q^{(i)}(s,a)$, joka kertoo kunkin toimintaprofiilin Q-arvon kussakin tilanteessa. Tämä edellyttää että kaikkien teot ovat muiden havaittavissa.
- Taulukon $Q^{(i)}$ esittäminen on vaikeaa, jos tiloja on monta. Tämä on ongelma myös yhden agentin tapauksessa, joten käytännössä tiloja joudutaan samastamaan toisiinsa vaikkapa joidenkin valittujen piirteiden perusteella tai hyödyntämällä esimerkiksi neuraaliverkkoja.
- Q-arvojen päivityksessä joudutaan uudessa tilassa etsimään maksimiarvoinen toimintaprofiili. Jos agenttien Q-arvoissa voi olla eroja, koordinointi voi tulla mahdottomaksi.
- Tutkiskelun ja hyödyntämisen välinen tasapainottelun täytyisi myös olla jotenkin koordinoitua.

Stokastiset nollasummapelit

- Myös kilpailutilanteissa voidaan yrittää käyttää tavanomaista Q-oppimista ja ajatella muut agentit vain ympäristöksi. Konvergenssi ei kuitenkaan ole taattua paitsi jos muut agentit käyttävät staattista politiikkaa. Parempi tapa on ottaa toisten agenttien toimet huomioon ja päivittää toimintaprofiileille yleistettyä Q-funktiota.
- Tarkastellaan yksinkertaista tapausta, jossa agentteja on kaksi ja ympäristönä stokastinen nollasummapeli.
- Stokastisissa peleissä eli Markov-peleissä pelitilanne muuttuu joka kierroksella stokastisesti edellisen pelitilanteen ja siinä tehtyjen toimintojen perusteella. Siis siirtymäfunktio $T : S \times A = (A_1, \ldots, A_n) \rightarrow PD(S)$ määrää seuraavan normaalimuotoisen pelin (joukossa S) todennäköisyyden.
- Stokastisten pelien luokka on toisaalta iteroitujen pelien yleistys satunnaistetuilla lopputuloksilla ja toisaalta Markov-päätösprosessien yleistys monelle agentille.

Minimax-Q

- Tällaisessa tilanteessa voidaan käyttää oppimisalgoritmia nimeltä *Minimax-Q*:
- Agentti i käyttää päivityssääntöjä

$$Q_i(s,a_1,a_2) \leftarrow (1-\lambda)Q_i(s,a_1,a_2) + \lambda(R_i(s,a_1,a_2) + \gamma V(s')),$$

$$V(s) \leftarrow \max_{P_1 \in PD(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} P_1(a_1) Q_1(s,a_1,a_2).$$

  Tässä siis ajatellaan vastapelaajan valitsevan omalta kannaltaan parhaan vaihtoehdon joten pyritään itse valitsemaan minimiä maksimoiva sekastrategia P_1.
- Algoritmi konvergoituu, mutta ei ole Hannan-konsistentti, koska jos toinen agentti ei pelaa optimaalista strategiaa, algoritmi ei osaa käyttää sitä hyväkseen.

Algoritmi

```
Initialize:
For all s in S, a in A, and o in O,
  Let Q[s,a,o] := 1
For all s in S,
  Let V[s] := 1
For all s in S, a in A,
  Let pi[s,a] := 1/|A|
Let alpha := 1.0
Choose an action:
With probability explor, return an action uniformly at random.
Otherwise, if current state is s,
  Return action a with probability pi[s,a].
Learn:
After receiving reward rew for moving from state s to s'
      via action a and opponent's action o,
  Let Q[s,a,o] := (1-alpha) * Q[s,a,o] + alpha * (rew + gamma * V[s'])
  Use linear programming to find pi[s,.] such that:
    pi[s,.] := argmax{pi'[s,.], min{o', sum{a', pi[s,a'] * Q[s,a',o']}}}
  Let V[s] := min{o', sum{a', pi[s,a'] * Q[s,a',o']}}
  Let alpha := alpha * decay
```

(Littman 1994, Fig. 1)

Yleiset stokastiset pelit

- Yleisissä stokastisissa peleissä on pyritty siihen, että oppiminen konvergoituisi Nash-tasapainoon.
- Tunnetuin algoritmi on *Nash-Q*, jossa agentti pyrkii arvioimaan toisten agenttien Q-arvoja ja päivittää omiaan sillä oletuksella että toiset käyttävät tasapainostrategiaansa.
- Konvergenssituloksia on saavutettu, mutta toistaiseksi vain rajoittavien teknisten oletusten vallitessa.
- Ongelmallinen kun Nash-tasapainoja on useita.

Uskomuksiin perustuva oppiminen

- Agentit voivat myös ylläpitää eksplisiittisesti uskomuksia toisten agenttien käyttämistä strategioista.
- Kaksi tunnettua oppimistekniikkaa joita käytetään toistetuissa (deterministisissä) peleissä: *fiktiivinen pelaaminen* (fictitious play) ja *bayesiläinen oppiminen* (Bayesian learning)
- Toimintaperiaate on seuraavanlainen:
  1. Ennen kutakin pelikierrosta agentilla on tietty arvio (uskomus) vastustajan strategiasta.
  2. Pelaaja pelaa sitä strategiaa, joka on paras vastaus vastustajan arvioituun strategiaan.
  3. Agentti päivittää uskomuksiaan vastustajan havaitun toiminnan perusteella.
- Menetelmät eroavat siinä kuinka uskomuksia ylläpidetään ja päivitetään.

Fiktiivinen pelaaminen

- Agentti pitää muistissa vastustajan aikaisempia valintoja ja pitää niiden empiiristä jakaumaa vastustajan käyttämänä sekastrategiana.
- Siis kullekin vastustajan teolle $a \in A$ jos vastustaja on valinnut a:n tähän mennessä $w(a)$ kertaa, niin oletetussa sekastrategiassa teon todennäköisyys on

$$P(a) = \frac{w(a)}{\sum_{a' \in A} w(a')}$$

- Pelaaja vastaa omalla parhaalla vastauksellaan. Erilaisia versioita saadaan käyttämällä erilaisia sääntöjä strategian valinnassa siinä tapauksessa että parhaita vastausstrategioita on useita.
- Konvergoituminen ei ole taattua, mutta jos empiirinen jakauma konvergoituu, se konvergoituu Nash-tasapainoon.

Bayesiläinen oppiminen

- Bayesiläisessä (tai rationaalisessa) oppimisessa agentille i attribuoitu strategiajoukko S_i voi sisältää puhtaiden ja sekastrategioiden lisäksi iteroitujen pelien strategioita (kuten Tit-for-Tat).
- Aluksi vastustajan strategioille asetetaan prioritodennäköisyydet.
- Todennäköisyyksiä päivitetään Bayesin säännöllä havaitun käyttäytymisen perusteella: posterioritodennäköisyys sille, että i käyttää strategiaa $s \in S_i$ kun havaittu historia on h, on

$$P_i(s|h) = \frac{P_i(h|s)P_i(s)}{\sum_{s' \in S} P_i(h|s')P_i(s')}$$

- Self-play-tilanteissa tietyillä ehdoilla agentit päätyvät lähes oikeisiin uskomuksiin toisen strategian havaittavasta osasta.
- Lisäksi self-play-tilanteissa tietyillä ehdoilla agenttien strategiat konvergoituvat suurella todennäköisyydellä kohti Nash-tasapainoa.

Lähteitä

- R.S. Sutton & A. Barto: *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- Vlassis, luku 8; Weiss, luku 6.
- Y. Shoham & K. Leyton-Brown: *Introduction to Multi-Agent Systems*.
- Minimax-Q: Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, 157–163.
- Nash-Q: J. Hu & M.P. Wellman: Nash-Q learning for general-sum stochastic games. *Journal of Machine Learning Research* 4, 2003, 1039–1069.
- Shoham, Y., Powers, R., Grenager, T., On the Agenda(s) of Research on Multi-Agent Learning. Artificial Multiagent Learning: Papers from the AAAI Fall Symposium 2004. Technical Report FS-04-02, The AAAI Press, 2004, 89-95.