SBNZ predlog projekta – Poker bot

Student: Danijel Radulović

Broj indeksa: SW-23/2016

Motivacija

Poker, odnosno igra kartama, je “igra nesavršenog znanja”, u kojoj učestvuje više takmičara, koji moraju razumeti procenu, predviđanje, upravljanje rizikom i čitanje protivničke igre, samim tim, predstavlja dobar primer sistema, u kome je potrebno doneti odluke u uslovima nesigurnosti i sistema, koji bi se mogao modelovati određenim pravilima.

Pregled problema

Problem, koji se rešava ovim projektom, predstavlja implementaciju bota, koji bi se mogao koristiti za igranje pokera. Njegovo ponašanje u različitim situacijama, u kojima može da se nađe tokom igre, bilo bi modelovano pravilima, zasnovanim na ekspertskom znanju. Tokom prethodnih dvadeset godina, pravljeni su poker botovi primenjujući razne metode, kao što su stabla igre, Monte Karlo simulacija, duboke neuronske mreže itd. Konkretno, performanse ovog bota bi se testirale pomoću PokerAcademy softvera, koji predstavlja platformu, na kojoj poker botovi mogu da igraju partije između sebe (<https://code.google.com/archive/p/opentestbed/source/default/source>), a kao literaturu bih koristio master rad sa Univerziteta u Alberti, na temu Dealing with Imperfect Information in Poker (<http://poker.cs.ualberta.ca/publications/papp.msc.pdf>).  
  
Metodologija rada

*Ulazi:* Vrednosti ulaza variraju na početku, a i u toku svake „ruke“ (partije) tačnije pri svakom menjanju faze jedne partije. U ulaze spadaju faza igre (*pre-flop, flop, turn, river*), broj igrača, pozicija bota za stolom, *hole cards* (karte koje je bot dobio), *board* (karte, koje su izašle), *pot* (suma novca ili čipova, koja se nalazi u igri), *bet/raise* (suma novca ili čipova, koje bi bot trebao da plati, ukoliko želi da nastavi igru), *blind* (suma, koju na početku partije moraju da plate dva igrača, zavisno od pozicije za stolom). Na osnovu ovih ulaza, bi se računale razne heuristike, koje bi se dalje koristile u pravilima.

*Izlazi:* Akcija, koju bi bot trebalo da sprovede

U prilogu je jednostavan primer, ukoliko je ukupno dva igrača i bot ima dobru „ruku“ (dobre početne karte).

Konkretan primer:

*Ulazi:* Faza *flop* (3 karte na terenu), ukupno dva igrača računajući i bota, prvi igrač je *check*-ovao, *hole cards* su *As Kh*, na *board*-u se nalaze *Ah 6s 9c*, *pot* 8, *blind* 4.

*Izlazi:* Akcija - *bet*, suma - 12 (3\**blind*)

Primer baze znanja: ulančana pravila

*Top pair* (*As Ah)* -> treba da igraš dalje, tj. da *call*-uješ/*bet-*uješ

Protivnik je *check*-ovao -> treba da *bet*-uješ

Faza *flop* -> *bet* 3\**blind*

Pravila implementirana u projektu

*Preflop rules* grupa pravila - Radna memorija sadrži PlayerState (trenutno stanje bota), kao globalne promenljive prosleđeni su razni *threshold* parametri, broj igrača koji bi još trebalo da odigraju tekuću ruku i podaci o igri (*GameInfo*)

1. Determine suits

*When*

Postoji PlayerState, čije karte su iste boje

*Then*

Postavlja se *sameSuit* na true

1. Starting hand rank - number of players is 5+, cards have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača u igri veći od 5 i karte su iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 5+ igrača i uslova da su karte iste boje

1. Starting hand rank - number of players is 5+, cards do not have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača u igri veći od 5 i karte nisu iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 5+ igrača i uslova da karte nisu iste boje

1. Starting hand rank - number of players is 3 or 4, cards have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača u igri 3 ili 4 i karte su iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 3-4 igrača i uslova da su karte iste boje

1. Starting hand rank - number of players is 3 or 4, cards do not have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača u igri 3 ili 4 i karte nisu iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 3-4 igrača i uslova da karte nisu iste boje

1. Starting hand rank - number of players is 1 or 2, cards have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača 1 ili 2 i karte su iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 1-2 igrača i uslova da su karte iste boje

1. Starting hand rank - number of players is 1 or 2, cards do not have same suit

*When*

Postoji PlayerState, gde je očekivani broj igrača 1 ili 2 i karte nisu iste boje

*Then*

Postavlja se odgovarajuća jačina početnih karata, na osnovu *IncomeRates* tabele za 1-2 igrača i uslova da karte nisu iste boje

1. Determine thresholds - tight and 1-2

*When*

Postoji PlayerState, gde je očekivani broj igrača 1 ili 2 i stil igre je *TIGHT*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive na osnovu formula, zasnovanih na ekspertskom znanju

1. Determine thresholds - tight and 3-4

*When*

Postoji PlayerState, gde je očekivani broj igrača 3 ili 4 i stil igre je *TIGHT*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Determine thresholds - tight and 5+

*When*

Postoji PlayerState, gde je očekivani broj igrača veći od 5 i stil igre je *TIGHT*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Determine thresholds - moderate and 1-2

*When*

Postoji PlayerState, gde je očekivani broj igrača 1 ili 2 i stil igre je *MODERATE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive

1. Determine thresholds - moderate and 3-4

*When*

Postoji PlayerState, gde je očekivani broj igrača 3 ili 4 i stil igre je *MODERATE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Determine thresholds - moderate and 5+

*When*

Postoji PlayerState, gde je očekivani broj igrača veći od 5 i stil igre je *MODERATE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Determine thresholds - loose and 1-2

*When*

Postoji PlayerState, gde je očekivani broj igrača 1 ili 2 i stil igre je *LOOSE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive

1. Determine thresholds - loose and 3-4

*When*

Postoji PlayerState, gde je očekivani broj igrača 3 ili 4 i stil igre je *LOOSE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Determine thresholds - loose and 5+

*When*

Postoji PlayerState, gde je očekivani broj igrača veći od 5 i stil igre je *LOOSE*

*Then*

Računaju se i postavljaju vrednosti za globalne *threshold* promenljive, vodeći računa i o činjenici da li je trenutna pozicija bot-a *small blind*

1. Strategy MAKE4

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju MAKE4

*Then*

Postavlja se strategija MAKE4

1. Strategy MAKE2

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju MAKE2

*Then*

Postavlja se strategija MAKE2

1. Strategy CALL2

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju CALL2

*Then*

Postavlja se strategija CALL2

1. Strategy MAKE1

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju MAKE1

*Then*

Postavlja se strategija MAKE1

1. Strategy CALL1

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju CALL1

*Then*

Postavlja se strategija CALL1

1. Strategy MAKE0

*When*

Postoji PlayerState, čija strategija nije nijedna od ekspertski definisanih strategija i jačina početnih karata je veća ili jednaka od *threshold*-a za strategiju MAKE0

*Then*

Postavlja se strategija MAKE0

1. Determine action - strategy MAKE4

*When*

Postoji PlayerState, čija je strategija MAKE4

*Then*

Postavlja se akcija *raise za 4 \* trenutna veličina bet-a*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy MAKE2

*When*

Postoji PlayerState, čija je strategija MAKE2

*Then*

Ako je broj trenutnih *raise*-ova, postavlja se akcija *raise za 2 \* trenutna veličina bet-a*, u suprotnom akcija *call*, koju bi bot trebalo da izvrši

1. Determine action - strategy CALL2

*When*

Postoji PlayerState, čija je strategija CALL2

*Then*

Postavlja se akcija *call za trenutni iznos*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy MAKE1 - scenario 1

*When*

Postoji PlayerState, čija je strategija MAKE1 i iznos koji bi trebalo *call*-ovati je veći ili jednak dvostrukoj trenutnoj veličini *bet-*a

*Then*

Postavlja se akcija *fold*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy MAKE1 - scenario 2

*When*

Postoji PlayerState, čija je strategija MAKE1 i iznos koji bi trebalo *call*-ovati je manji od dvostruke trenutne veličine *bet-*a

*Then*

Postavlja se akcija *call za trenutni iznos*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy MAKE1 - scenario 3

*When*

Postoji PlayerState, čija je strategija MAKE1 i broj igrača, koji su *bet/raise* ovu ruku je jednak nuli

*Then*

Postavlja se akcija *raise za trenutnu veličinu bet-*a, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy CALL1 - scenario 1

*When*

Postoji PlayerState, čija je strategija CALL1 i iznos koji bi trebalo *call*-ovati je veći ili jednak dvostrukoj trenutnoj veličini *bet-*a

*Then*

Postavlja se akcija *fold*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy CALL1 - scenario 2

*When*

Postoji PlayerState, čija je strategija CALL1 i iznos koji bi trebalo *call*-ovati je manji od dvostruke trenutne veličine *bet-*a

*Then*

Postavlja se akcija *call za trenutni iznos*, kao akcija, koju bi bot trebalo da izvrši

1. Determine action - strategy MAKE0

*When*

Postoji PlayerState, čija je strategija MAKE0

*Then*

Ako je trenutni iznos, koji bi trebalo *call*-ovati veći od 0, postavlja se akcija *fold*, u suprotnom, akcija *check*, kao akcija, koju bi bot trebalo da izvrši