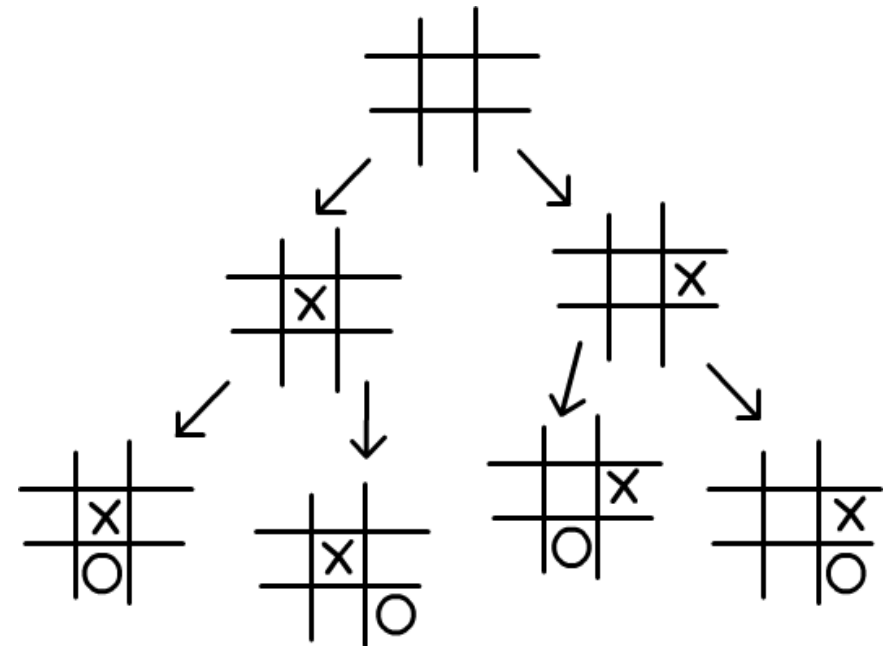
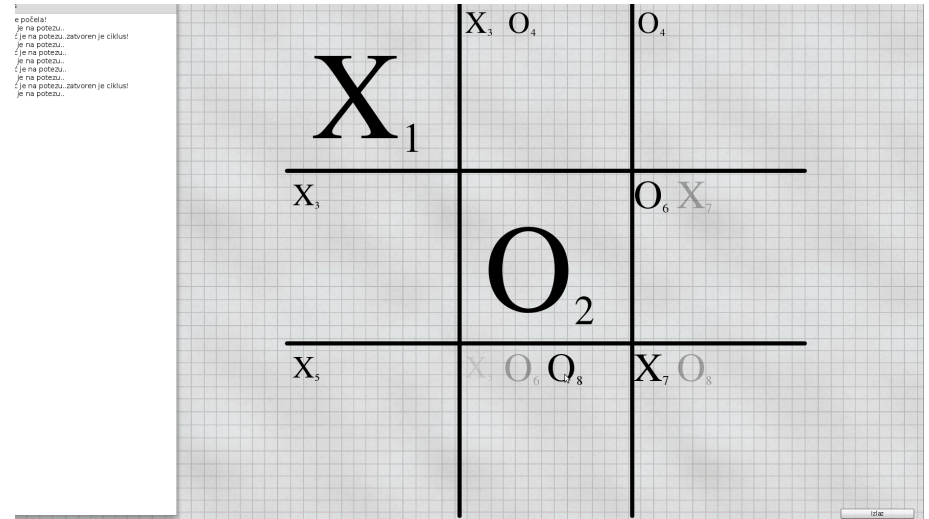


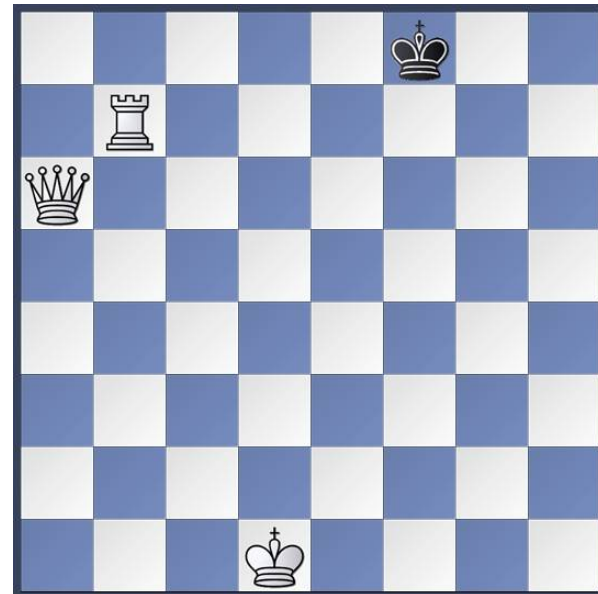
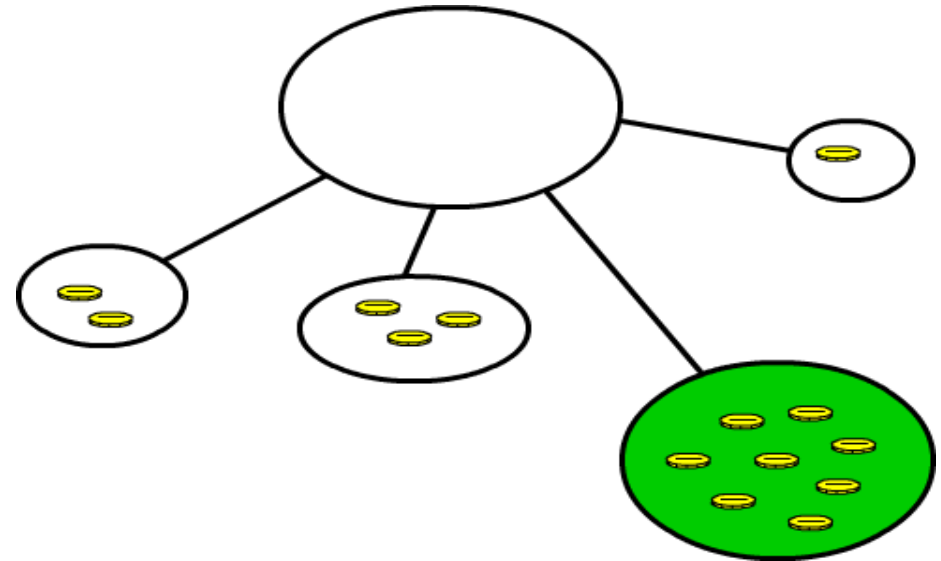
Uvod

- Kvantni križić kružić – varijanta klasične igre u kojoj igramo više klasičnih igara paralelno I posebnim potezom odlučujemo koja varijanta se “stvarno dogodila”
- više o igri na Wikipediji (Quantum Tic-Tac-Toe)
- postojeća implementacija radi min-max s alfa-beta podrezivanjem
- stablo igranja se širi eksponencijalno, te zbog toga ne možemo duboko tražiti
- umjesto toga, želimo računalnog igrača učiti nekom metodom strojnog učenja
- Uspješnost igranja možemo lako vidjeti (broj dobivenih partija npr.), a izvor podataka su odigrane partije



Ciljna funkcija i skup stanja

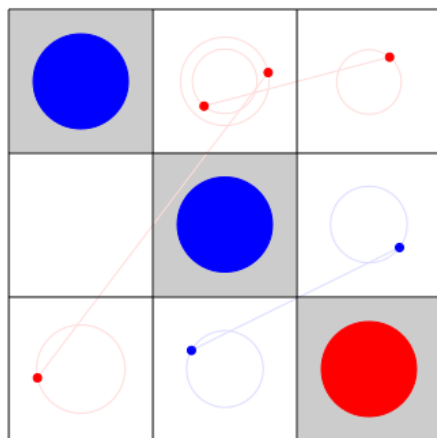
- Izbor slijedećeg poteza ovisi isključivo o trenutnom stanju
- učenje funkcije koja pridružuje brojčanu vrijednost određenom stanju (umjesto najboljeg slijedećeg poteza) nam daje veću fleksibilnost
- $f: \mathbf{S} \rightarrow \mathcal{R}$, gdje je \mathbf{S} skup svih legalnih stanja ploče, tako da $f(a) \leq f(b)$ akko {stanje a je bolje ili jednako stanju b u smislu vjerojatnosti da križić pobijedi}
- uočimo da reprezentacija stanja ne mora sadržavati pozicije znakova na ploči – bilo bi idealno kada bi postojali nekakvi “definirajući atributi” - time bismo značajno smanjili domenu hipoteza
- no, nemamo dovoljno znanja o igri da bi znali koji su to (i postoje li)



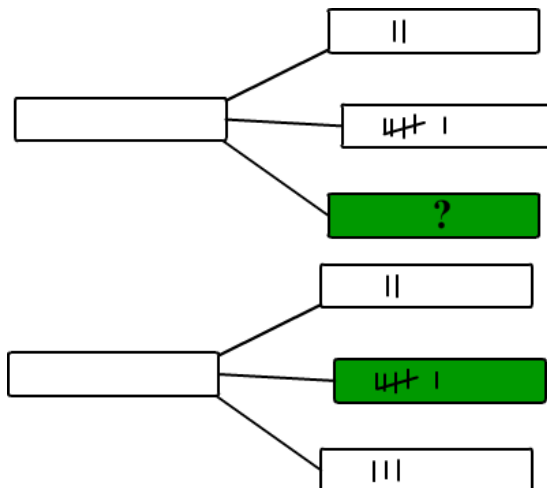
Dosadašnja istraživanja

QTPy

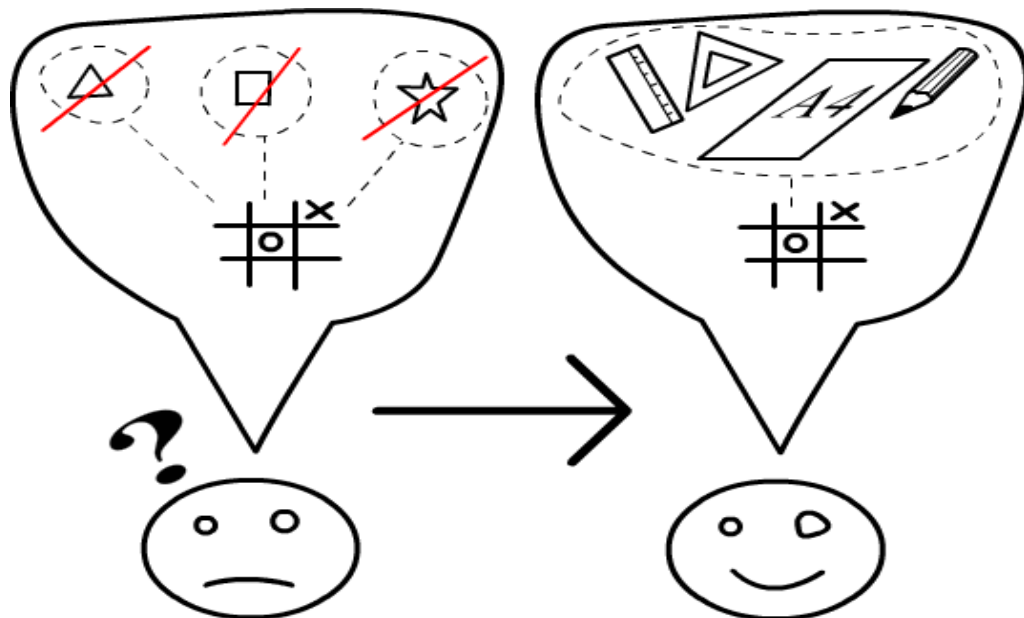
⊙ = ⊙



Your turn (1 of 2)

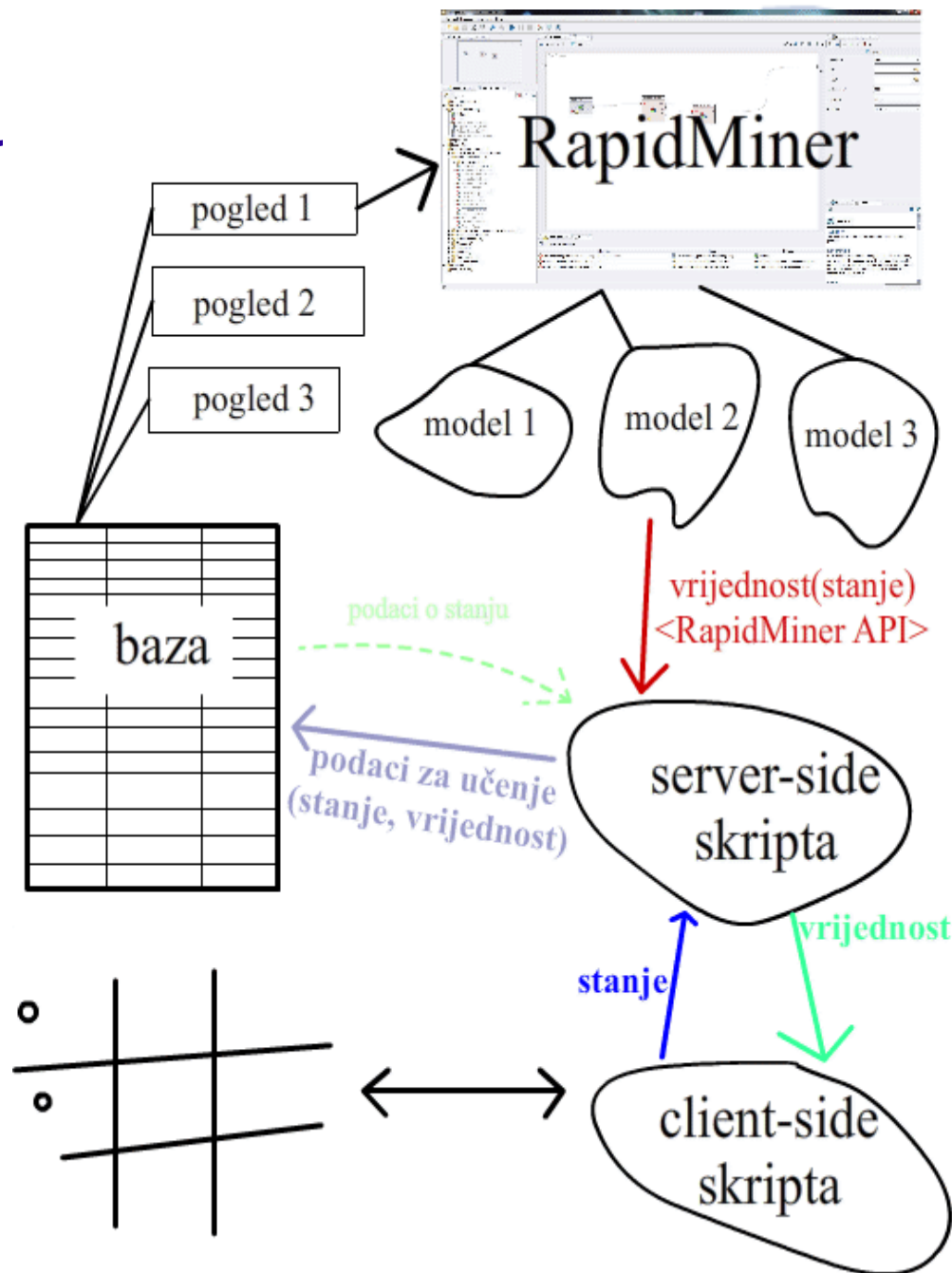


- Implementacija na <http://www.qtpy.mobi/>
- vraća slijedeći potez, kao ulazni podatak dobiva pozicije figura na ploči
- podaci su oblika [stanje, potez, odigrano, neriješeno, izgubljeno]
- vraćaju se neodigrani potezi ili oni koji su najčešće sudjelovali u pobjedi
- očita zamjerka je relativno slaba igra
- igra i primjena strojnog učenja na nju su relativno neistraženi
- nije sasvim jasno kako bi reprezentacija ciljne funkcije i prostor hipoteza trebali izgledati
- umjesto fiksnog algoritma, napraviti ćemo okruženje u kome relativno jednostavno možemo testirati razne metode



Plan istraživanja

- svaki puta kada je računalni igrač na potezu, šalje serveru upite za sva stanja do kojih može doći legalnim potezom
- u upitu šalje metapodatke o stanjima (fiksirane pozicije, tko je na potezu) itd.
- od servera dobiva brojčanu vrijednost za svako stanje, i njegove metapodatke (koliko često je igrano npr.) – s obzirom na to odlučuje koji će potez odigrati
- sa strane servera je skripta koja izvrednjava stanje pomoću nekog modela iz RapidMinera
- na kraju igre se sva njena stanja spremaju u bazu (podaci za učenje)



Rezultati projekta

- online verzija igra u kojoj možemo igrati protiv računala (koje igra “dovoljno dobro” odn. dovoljno je jako da bude zanimljivo)
- generator eksperimenata s raznim računalnim igračima i statistika
- istraživanje kvantnog križić kružića i primjene metoda strojnog učenja na njega
- upoznavanje RapidMiner APIa i uklapanje istog u složenije softversko okruženje
- primjena metoda strojnog učenja u web aplikacijama

The screenshot shows a web application with three main sections: a game board, a settings panel, and a results table.

Game Board: A 3x3 grid with 'X' marks at (1,1), (1,3), (2,2), and (3,2), and an 'O' mark at (3,1). Navigation buttons 'Igra', 'Generator eksperimenata', and 'Statistika' are at the top.

Settings Panel: Contains two identical blocks for configuring experiments. Each block has dropdowns for 'No. kuglastih', 'No. kuglastih', and 'No. kuglastih', and checkboxes for 'Koristi sirove podatke o pozicijama', 'Koristi podatke o redosledu broja poteza', and 'Koristi metapodatke o otpisanim poljima'. A 'Generiraj' button is at the bottom.

Results Table: A table showing the results of 3000 games.

	Dobiveno	Izgubljeno	Neriješeno
Ukupno	432	781	200
Ljudski igrači	112	416	34
Rote (:raw => true, ..)	231	36	1
Rote (:raw => false, ..)	2	108	0