

Tema de casa 2 - Clasificare

Valentin Lungu

Laboratorul AIMAS,
Facultatea de Automatica si Calculatoare

5 Aprilie 2014

Data afisarii: 5.04.2014

Termen predare: 24.04.2014

Data modificari: 21.04.2014

Se accepta maxim 4 zile intarziere, cu penalizare (din 10 puncte) astfel:

- 0.5 puncte in prima zi;
- 1punct /zi pentru urmatoarele 3 zile.

Tema se prezinta la laborator.

Limbaje de programare: C, C++, C#, Java, Scheme, Prolog, Lisp, Haskell, Pyhton*

*se accepta si alte limbaje, dar intrebati inainte

Enunt

Fie urmatorul joc. Intr-un mediu bidimensional (spatiu continuu, nu grid/matrice) exista un animal prada, mai multi pradatori si o serie de capcane. Aceste animale au trei caracteristici: raza de perceptie, raza de coliziune si viteza cu care se deplaseaza. Capcanele au doar raza de influenta. Creaturile din acest habitat se supun urmatoarelor reguli:

- un animal care trece printr-o capcana este incapacitat permanent (iese din joc)
- doar doi sau mai multi pradatori pot dobori prada (cand prada este prinsa de pradatori, jocul este pierdut)
- prada poate incapacita un pradator prin contact direct, atata timp cat coliziunea este doar cu acesta, sau prin atragerea lor in capcane
- fiecare creatura este constienta doar de elementele aflate in campul sau de perceptie
- pradatorii vor alerga dupa prada daca aceasta este in campul vizual si nu vor ocoli capcanele
- prada va sta pe loc daca nu este amenintata (nu are pradatori in campul vizual), sau va incerca sa se salveze altfel
- modificarile directiei de mers se produc instantaneu
- **pradatorii isi vor sincroniza vitezele intre ei astfel incat sa ajunga prada in acelasi timp (daca se vad)**
- **daca un pradator se gaseste singur cu prada, el va incerca sa o doboare (si va esua)**
- **daca pradatorii nu au prada in raza lor vizuala, acestia vor incerca s-o caute**

Cerinte

Se cere dezvoltarea unui sistem cu urmatoarele componente:

- **modul preprocesare** - acest modul extrage caracteristicile importante ale starii sistemului (zona de perceptie a prazii)
- **modul clasificare (ex. k-means)** - rolul acestui modul este de a determina si reduce numarul starilor stabile ale sistemului
- **modul recunoastere (ex. retea neurala)** - rolul acestui modul este de a aduce starea sistemului intr-una din starile stabile (repararea starii)
- **modul invatare prin recompensa (ex Q-learning)** - are rolul de a invata cea mai buna actiune (in cazul nostru directie de mers) pentru fiecare din starile stabile

Sistemul functioneaza in doua etape (modulul de preprocesare este folosit pentru a extrage informatiile utile de fiecare data):

- **clasificare**
 - jocul este rulat cu miscarea aleatoare a prazii pentru a determina starile stabile. Numarul de stari stabile este ales de dezvoltator. Folositi cate considerati necesare.
- **invatare** (in aceasta etapa, prada va urma indicatiile sistemului)
 - dupa determinarea starilor stabile, modulul de recunoastere este antrenat cu acestea
 - algoritmul de invatare este folosit pentru a asocia o directie de deplasare cu fiecare stare

Jocul si algoritmul se ruleaza pana cand prada castiga un procent satisfactor de jocuri (alegerea dezvoltatorului).

Se cer:

- **[1p]** implementarea jocului
 - numarul de pradatori si dimensiunea habitatului sa poata fi variat
- **[8p]** implementarea sistemului
 - **[2p]** modul clasificare
 - **[3p]** modul recunoastere
 - **[2p]** modul invatare
- **[1p]** implementarea unei interfete grafice pentru a putea urmari actiunea (nu este necesar ceva avansat; o serie de cercuri cu vectori viteza pe o plansa sunt suficiente)
- **[1p]** calculul procentului de jocuri castigate pe batch (jocurile vor fi rulate in batchuri de 100)
- **[2p]** bonus
- parametrii jocului vor fi cititi dintr-un fisier cu urmatorul format:

```
dimensiune_habitat_1
dimensiune_habitat_2
viteza_prada
raza_coliziune_prada
raza_perceptie_prada
numar_pradatori
viteza_pradatori
raza_coliziune_pradatori
numar_capcane
raza_coliziune_capcane
```

Se recomanda:

- posibilitatea varierii parametrilor principali ai tehnicilor folosite (numar de neuroni, numar de straturi, numar de clustere, etc.)

Observatii:

- actiunile de atac nu trebuie executate, ele se intampla cand este detectata o coliziune
- la prezentarea temei va trebui sa justificati deciziile de proiectare luate (alegerea tehnicilor, parametrilor, etc.)
- pentru testare puteti simula o haita de **4-11** pradatori urmarind o prada
- pozitiile initiale ale prazii, pradatorilor si capcanelor vor fi aleatoare in habitat (fara a exista coliziuni initiale)
- coliziunile sunt calculate intre razele de coliziune a doua entitati

Bonus

Se adauga urmatoarele reguli:

- pradatorii vor urmari prada doar daca sunt doi sau mai multi in campul lor vizual (daca vad si alti pradatori indreptandu-se spre prada)
- pradatorii se vor indrepta unii spre altii atata timp cat nu urmaresc prada (pradatorii stiu directia generala spre ceilalti pradatori)