Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Il Ritorno dell'Ubriaco
(Da vedere insieme)

Esercizio: Il Ritorno dell'Ubriaco

Dopo una serata di bagordi, Gigi deve tornare a casa a piedi

- All'inizio, Gigi si muove in una data direzione (verso casa)
- Ad ogni passo, Gigi mantiene la direzione con probabilità p
- Oppure cambia direzione, con probabilità $1 - p$

Intuitivamente p controlla il livello di sobrietà

- Con $p = 1$ Gigi è sobrio: cammina sicuro verso casa
- Con $p = 0.5$ Gigi arrivare a casa è questione di pura fortuna

Vogliamo osservare il moto di Gigi nel tempo

- Lo faremo modellando il moto come un sistema dinamico
- Usiamo un approccio "stocastico" (numeri pseudo-casuali)

Esercizio: Il Ritorno dell'Ubriaco

Cosa è lo stato $x^{(t)}$ per il nostro problema?

Lo $X^{(t)}$ deve contenere abbastanza informazioni per determinare $X^{(t+1)}$

- Quindi ci serve la posizione di Gigi
- Ma anche la direzione dell'ultimo movimento!

Diciamo che lo stato è un vettore di due componenti:

$$X = (x, d)$$

- Convenzione: in questo esercizio X è lo stato e x la posizione
- Va capito come rappresentare la direzione con un numero

Con queste due informazioni possiamo determinare lo stato futuro

Esercizio: Il Ritorno dell'Ubriaco

A partire dal file `es_drunkman.m` nello start-kit:

Definite la funzione di transizione:

```
function xf = f(xc)
```

- HP: `xc(1)` = posizione, `xc(2)` direzione dell'ultimo spostamento

Il codice di simulazione è già disponibile nella funzione principale

- Visualizzate come si evolve lo stato nel tempo
- Ci sono due comportamenti ⇒ ci servono due `plot`
- Potete usare `hold on` per evitare la sovrascrittura dei disegni
- Cambiate lo stato si sobrietà p e vedete come varia il moto
- Per quali valori di p il moto sembra più verosimile?

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Un Modello Preda-Predatore

Esercizio: Un Modello Preda-Predatore

I modelli preda-predatore:

- Rappresentano la co-evoluzione di due popolazioni antagoniste
- E.g. prede-predatori, ma anche reazioni chimiche che si ostacolano

Tipicamente ci sono due elementi nel vettore dello stato:

- Il numero di "prede" H
- Il numero di "predatori" P

Simulando il sistema:

- Si può studiare il comportamento delle due popolazioni
- Si può verificare se vi sia un equilibrio stabile

Esercizio: Un Modello Preda-Predatore

Consideriamo questo modello preda-predatore:

$$H^{(t+1)} = \overbrace{r\left(1 - \frac{H^{(t)}}{k}\right) H^{(t)}}^{\text{crescita logistica}} - \overbrace{s\, H^{(t)} P^{(t)}}^{\text{prede eliminate}}$$

$$P^{(t+1)} = \underbrace{u\, P^{(t)}}_{\text{calo in assenza di prede}} + v\underbrace{\left(s\, H^{(t)} P^{(t)}\right)}_{\text{prede eliminate}}$$

- k è la massima popolazione di prede sostenibile
- s è la frazione di $H^{(t)}$ che un predatore può "mangiare"
- $u < 1$ è il ritmo di scomparsa dei predatori in assenza di prede
- v è il "bonus riproduttivo" per ogni preda "mangiata"

Esercizio: Un Modello Preda-Predatore

Nello start-kit, aprire e modificare il file `es_log_pp.m`

Implementare la funzione (ausiliaria) di transizione `f(xc)`:

- HP: `xc(1)` sia il numero di prede e `xc(2)` quello di predatori
- I valori di r, k, s, u, v sono già definiti nel codice

Il codice di simulazione è già disponibile nella funzione principale:

- Lo stato iniziale si assume sia $(H, P) = (100, 30)$
- Disegnate l'andamento delle due popolazioni nel tempo
  - Potete usare due `plot` (e `hold on` per non sovrascrivere)
- Disegnate la traiettoria nello spazio degli stati
  - Potete usare `plot` (la coordinata x è H, y è P)

Esercizio: Un Modello Preda-Predatore

Provate a fare delle variazioni!

Aumentate e diminuite (solo) s:

- Cosa vi aspettate che succeda? Cosa succede?

Aumentate e diminuite (solo) r:

- Cosa vi aspettate che succeda? Cosa succede?

Aumentate e diminuite (solo) u:

- Cosa vi aspettate che succeda? Cosa succede?

Riuscite a spiegare intuitivamente le ragioni?

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Scorte di Magazzino

Esercizio: Scorte di Magazzino

Si deve analizzare il livello di scorte di un magazzino

- Il magazzino ha una capacità di 10 unità
- Inizialmente, il magazzino è pieno
- Ogni giorno, c'è una probabilità del 10% che una unità sia venduta
- Il magazzino non può contenere meno di 0 unità
- Il magazzino viene riempito la sera di ogni 15° giorno

Si desidera studiare l'andamento delle scorte

Un possibile approccio:

- Definiamo un modello stocastico (basato su numeri pseudo-casuali)
- Effettuiamo qualche simulazione

Esercizio: Scorte di Magazzino

Come base, utilizzate il file `es_warehouse_stoch.m` nello start-kit

La funzione (ausiliaria) di transizione ha come interfaccia:

```
function xf = f(xc, t)
```

- **`xc`** è il livello delle scorte corrente
- Lo stato futuro dipende anche dal tempo **`t`**!
- Se **`t`** è un multiplo di 15, lo stato futuro sarà 10

Il codice di simulazione è già disponibile nella funzione principale:

- Visualizzate lo stato nel tempo, per un periodo sufficientemente lungo
- Visualizzate il numero di intervalli passati in ogni stato
- Per la seconda domanda potete usare **`histogram`**!

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Livello di Energia di una Elettrone

Esercizio: Livello di Energia di un Elettrone

Un elettrone e può trovarsi a tre livelli di energia diversi

- e si trova inizialmente a livello 1 (il più basso)
- e è soggetto ad interazioni che possono cambiare il livello di energia
- Sappiamo che lo stato di e è ben descritto in termini di probabilità

Vogliamo determinare la probabilità che e sia in ciascun livello

Un possibile approccio:

- Modelliamo la particella come un sistema dinamico tempo-discreto
- Definiamo un modello stocastico (numeri pseudo-casuali)
- Simuliamo ed analizziamo l'istogramma dello stato

In questo modo approssimiamo la distribuzione di probabilità

Esercizio: Livello di Energia di un Elettrone

Al livello 1:

- Con il 75% di probabilità, l'elettrone resta stabile
- Con il 25% di probabilità, l'elettrone si sposta al livello 2

Al livello 2:

- Con il 25% di probabilità, l'elettrone resta stabile
- Con il 50% di probabilità, l'elettrone si sposta al livello 1
- Con il 25% di probabilità, l'elettrone si sposta al livello 3

Al livello 3:

- Con il 25% di probabilità, l'elettrone resta stabile
- Con il 75% di probabilità, l'elettrone si sposta al livello 2

Esercizio: Livello di Energia di un Elettrone

Partite dal file `es_electron_stoch.m` nello start-kit

- Definite la funzione (ausiliaria) di transizione:

```
function xf = f(xc)
```

- Il codice di simulazione è già definito nella funzione principale
- Aggiungete il codice per visualizzare lo stato nel tempo
- Aggiungete il codice per l'istogramma

Il codice di simulazione è già disponibile nella funzione principale:

- Modificate il numero di passi...
- ...Finché non vi accorgete che l'istogramma si stabilizza
- In questo modo avrete una buona approssimazione delle probabilità

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Modello di Shepherd

Esercizio: Modello di Shepherd

Il modello di Shepherd è una modifica di quelli di Beverton-Holt

Descrive l'andamento di una singola popolazione, con l'iterazione:

$$x^{(t+1)} = \frac{r\, x^{(t)}}{1 + \left(\frac{x^{(t)}}{k}\right)^2}$$

- r è il tasso di crescita
- k di popolazione che, se raggiunto, dimezza il tasso di crescita

Vogliamo analizzare il comportamento del modello

- Lo faremo implementando un simulatore
- Quindi disegnando grafici ed osservando i risultati

Esercizio: Modello di Shepherd

Partite dal file `es_shepherd.m` nello start-kit

Definite la funzione (ausiliaria) di transizione

```
function xf = f(xc)
```

- Assumete $r = 1.5, k = 1000$

Il codice di simulazione è già disponibile nella funzione principale

- Assumete che la popolazione iniziale sia di 100 unnità
- Visualizzate l'andamento dello stato nel tempo
- Osservate come varia lo stato finale al variare di r
- Determinate il minimo valore di r tale che la popolazione finale sia k