Laboratorio di Informatica T

Sistemi Dinamici Tempo Discreti

Attualità

Lunedì scorso, questi signori hanno vinto un premio Nobel

Michael Rosbash, Jeffrey C. Hall, and Michael W. Young

Ritmo Circadiano

Ma cosa hanno fatto?

Hanno identificato i meccanismi fondamentali dietro il ritmo circadiano

- Il ritmo circadiano è il nostro "orologio biologico"
- Ha un periodo di circa 24 ore (circa-diem)
- Può adattarsi a segnali esterni (e.g. luce)

Più in dettaglio, come funziona?

- Una proteina "X" si accumula durante la notte...
- ...E viene smaltita durante il giorno

Ma come fa ad invertirsi l'andamento? La reazione è autocatalitica

- La proteina "X" è contemporaneamente un reagente ed un prodotto
- La proteina "X" inibisce la propria produzione

Oscillatore di Van der Pol

Il ritmo circadiano è un esempio di oscillatore biologico

- Le vere reazioni che lo caratterizzano sono complesse…
- …Ma un modello semplificato è alla nostra portata!

Oscillatore di Van der Pol (discretizzato)

Consideriamo una versione discretizzata dell'oscillatore di Van der Pol:

$$x^{(t+1)} = x^{(t)} + hy^{(t)}$$
$$y^{(t+1)} = y^{(t)} + h\left(\mu(1 - x^{(t)2})y^{(t)} - x^{(t)}\right)$$

- x è la variabile che corrisponde alla proteina, y è ausiliaria
- Hanno un valore diverso per ogni istante di tempo t
- Le equazioni ci dicono come passare dallo stato al tempo t…
- …Allo stato al tempo $t + 1$

Sistemi Dinamici Tempo Discreti

Si dice sistema dinamico un sistema che ha uno stato che varia nel tempo

- Nel nostro caso, lo stato sono le due variabili x e y…
- …Che variano nel tempo per passi discreti

Per questa ragione, il nostro sistema dinamico è tempo-discreto

- In generale, un sistema dinamico tempo discreto…
- …È caratterizzato dall'equazione:

$$x^{(t+1)} = f(x^{(t)})$$

- x può essere un vettore (i.e. molte variabili)
- f si dice funzione di transizione

Simulazione di Sistemi Tempo-Discreti

Torniamo al nostro oscillatore:

$$x^{(t+1)} = x^{(t)} + hy^{(t)}$$
$$y^{(t+1)} = y^{(t)} + h\left(\mu(1 - x^{(t)2})y^{(t)} - x^{(t)}\right)$$

- Se conosciamo h, μ ed i valori iniziali di x e y…
- …Possiamo simulare l'andamento dello stato

Lo pseudo-codice per il nostro algoritmo potrebbe essere:

$x_{cur} = x_0$
for $t = 1..n$
    $X_t = x_{cur}$
    $x_{cur} = f(x_{cur})$

- x_0 è il valore iniziale dello stato, x_{cur} quello corrente
- Alla fine, la matrice X contiene il risultato

Simulazione di Sistemi Tempo-Discreti

Per l'implementazione, iniziamo dalla funzione di transizione:

```
function Xf = f(X, h, mu)
    % "Spacchetto" lo stato
    x = X(1);
    y = X(2);
    % Calcolo lo stato futuro
    Xf(1) = x + h * y;
    Xf(2) = y + h * (mu * (1 - x^2)*y - x);
end
```

- `X` dovrà contenere lo stato corrente...
- ...Nel nostro caso, un vettore con x e y
- La funzione restituisce i due elementi dello stato futuro in `Xf`

Simulazione di Sistemi Tempo-Discreti

Quindi possiamo effettuare la simulazione in uno script:

Prima definiamo i dati del problema:

```
function es_van_der_pol()
    % Dati del problema
    x0 = 1;
    y0 = 1;
    h = 0.05;
    mu = 2;

    ...
end
```

Simulazione di Sistemi Tempo-Discreti

Quindi possiamo effettuare la simulazione in uno script:

Poi effettuiamo la simulazione vera e propria:

```
function es_van_der_pol()
  ...
  % Simulazione
  X = [];
  T = 1:1000; % Istanti di tempo
  xc = [x0, y0]; % Stato iniziale
  for t = T
      X(t, :) = xc; % Salvo lo stato corrente
      xc = f(xc, h, mu); % Genero lo stato futuro
  end
  ...
end
```

Simulazione di Sistemi Tempo-Discreti

Quindi possiamo effettuare la simulazione in uno script:

Infine disegniamo l'andamento delle grandezze che ci interessano

```
function es_van_der_pol()
  ...
  % "Spacchetto" le due componenti dello stato
  x = X(:, 1); % Prima colonna
  y = X(:, 2); % Prima colonna

  % Disegno l'andamento nel tempo
  plot(T, x)
end
```

Esercizio

Studiate il comportamento dell'oscillatore

Il codice è disponibile nello start-kit

- Cosa succede con i valori dei parametri suggeriti?
- Cosa succede variando x_0 e y_0?
- Cosa succede facendo crescere μ (in $[1, 6]$)?
- Cosa succede ponendo $\mu = 1$ e $h = 0.315$?
  - Suggerimento: simulate solo 300 iterazioni in questo caso

Esercizio

Risposte:

- Cosa succede con i valori dei parametri suggeriti?
  - Il sistema ha un comportamento periodico
- Cosa succede variando x_0 e y_0?
  - Niente! Il ciclo è robusto rispetto a variazioni dello stato iniziale
  - Si dice che il sistema ha un ciclo limite
- Cosa succede facendo crescere μ (in $[1, 6]$)?
  - Il ciclo diventa più "squadrato": l'oscillatore di Van der Pol...
  - ...è nato come un modello del battito cardiaco!
- Cosa succede ponendo $\mu = 1$ e $h = 0.315$?
  - Emergono delle irregolarità non predicibili
  - Il comportamento diventa caotico

Laboratorio di Informatica T

Esercizio: Modello di Ricker

Esercizio: Modello di Ricker

Modello di Ricker

È un sistema dinamico tempo-discreto caratterizzato dall'equazione:

$$x^{(t+1)} = x^{(t)} e^{r\left(1-\frac{x^{(t)}}{N}\right)}$$

È nato per modellare l'evoluzione di una popolazione di salmoni

- x è il numero di individui
- r è un tasso di crescita
- N è la popolazione sostenibile

Il sistema può essere simulato in modo analogo a quanto visto

- Questa volta lo stato è uno scalare...
- ...Il che rende le cose più semplici

Esercizio: Modello di Ricker

Il modello di Ricker può assumere una varietà di comportamenti:

Partite dal file **`es_ricker.m`** nello start-kit

- Definite la funzione (corrispondente alla funzione di transizione):

```
function xf = f(x, r, N)
```

- Nella funzione principale, simulate (e disegnate) 100 passi

Determinate per via empirica per quali valori di r:

- Il sistema converge ad uno stato stabile, senza oscillare
- Il sistema converge ad uno stato stabile, ma dopo aver oscillato
- Il sistema ha un comportamento periodico
- Il sistema ha un comportamento caotico

Laboratorio di Informatica T

Esercizio: Modello di Beverton-Holt

Esercizio: Modello di Beverton-Holt

Sia data una popolazione cresce secondo il modello

$$x^{(t+1)} = \frac{r\, x^{(t)}}{1 + \frac{x^{(t)}}{N}}$$

Dove:

- r indica il tasso di crescita (deve valere $r \in [1.0, 2.0]$)
- N indica un valore di popolazione…
- …che, se raggiunto, dimezza il tasso di crescita

Il modello è nato per applicazioni simili a quello di Ricker

Si implementi un simulatore per il modello

Esercizio: Modello di Beverton-Holt

In particolare, partite dal file `es_beverton_holt.m` nello start-kit:

Definite la funzione di transizione:

```
function xf = f(x, r, N)
```

- Che calcoli lo stato futuro **`xf`** a partire da quello corrente **`x`**
- Nella funzione principale, simulate (e disegnate) 100 passi

Studiate per via empirica il comportamento del sistema:

- Per quali valori di r la popolazione cresce?
- Per quali collassa?
- Il sistema può oscillare?
- Il sistema può assumere comportamento periodico?
- Il sistema può assumere comportamento caotico?

Laboratorio di Informatica T

Esercizio: Modello di Shepherd

Esercizio: Modello di Shepherd

Sia data una popolazione cresce secondo il modello

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

Dove:

- r indica il tasso di crescita (positivo)
- N indica un valore di popolazione...
- ...che, se raggiunto, dimezza il tasso di crescita

Il modello è nato per applicazioni simili a quello di Ricker

Si implementi un simulatore per il modello

Esercizio: Modello di Shepherd

In particolare, partite dal file `es_shepherd.m` nello start-kit:

Definite la funzione di transizione:

```
function xf = f(x, r, N)
```

- Che calcoli lo stato futuro **`xf`** a partire da quello corrente **`x`**
- Nella funzione principale, simulate (e disegnate) 100 passi

Studiate per via empirica il comportamento del sistema:

- Per quali valori di r la popolazione si azzera?
- Per quali si assesta sul valore N?
- Il sistema può oscillare?
- Il sistema può assumere comportamento periodico o caotico?

Laboratorio di Informatica T

Esercizio: Trasposizione di Matrice

Esercizio: Trasposizione di Matrice

Nel file di funzione **`es_transpose.m`** si definisca la funzione ausiliaria:

```
function B = my_transpose(A)
```

- Che calcoli la trasposizione della matrice **`A`**

Si verifichi la correttezza:

- Nella funzione principale **`es_transpose`**
- Utilizzando delle matrici di numeri casuali
- Confrontandosi con l'operatore di trasposizione di Matlab

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Conteggio di Elementi

Esercizio: Conteggio di Elementi

Nel file di funzione **`es_count.m`**, si definisca la funzione ausiliaria:

```
function [U, C] = my_count(V)
```

Che, dato un vettore di ingresso **`V`**, restituisca **`U`** e **`C`** tali che:

- **`U`** contenga gli elementi distinti di **`V`**
  - Suggerimento: li potete ottenere con **`unique`** di Matlab
- Per ogni valore **`v`** in **`U`**, la cella corrispondente di **`C`**…
- …Contenga il numero di occorrenze di **`v`** in **`V`**

I.e. la funzione deve contare le occorrenze di ogni elemento.

- Si verifichi il funzionamento nella funzione principale
- Si utilizzi un vettore definito a mano

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Prodotto Cumulativo

Esercizio: Prodotto Cumulativo

Matlab fornisce la funzione:

```
function P = cumprod(V)
```

- Che in ogni elemento `P(ii)` del vettore restituito...
- ...riporta il prodotto degli elementi di `V` negli indici da `1` a `ii`

Quindi, per esempio:

```
cumprod([2, 4, 6]) % denota [2, 8=2*4, 48=2*4*6]
```

Esercizio: Prodotto Cumulativo

Nel file di funzione **`es_cumprod.m`**, si definisca una funzione:

```
function P = my_cumprod(V)
```

- Che replichi il comportamento di **`cumprod`**

Si verifichi la correttezza nella funzione principale:

- Si utilizzino dei vettori di numeri casuali
- Si confrontino i risultati con quelli di **`cumprod`** in Matlab

Laboratorio di Informatica T

Esercizio: Linear Congruential Generator

Esercizio: Linear Congruential Generator

Si consideri la successione (linear congruential generator):

$$X_{n+1} = (aX_n + c) \bmod m$$

- Con X_i, a, c ed m interi; il valore di partenza X_0 è noto

Nel file di funzione **`es_lcg.m`** si definisca la funzione ausiliaria:

```
function R = my_lcg(X0, N)
```

- Che, dato il valore (scalare) di X_0…
- …Generi gli elementi da X_1 a X_N della successione…
- …E li salvi nel vettore di ritorno **`R`**

Si consideri $m = 16, a = 9, c = 3$

Esercizio: Linear Congruential Generator

Suggerimento: somiglia al codice per simulare sistemi dinamici!

- Ricordate che $x \mod y$ è il resto della divisione intera tra x e y
- In Matlab lo potete ottenere con `mod(x, y)`

Si effettuino dei test nella funzione principale `es_lcg`:

- Si provi ad eseguire `my_lcg` in sequenza con diversi valori di `X0`

Noterete che i valori di uscita variano in modo imprevedibile

- LCG può essere utilizzato per ottenere numeri pseudo-casuali...
- Ossia numeri che sembrano casuali, ma non lo sono
- La sequenza è controllata da X_0, che si dice anche "seme" (seed)

La funzione `rand` in Matlab funziona in modo simile! Ci ritorneremo su...