

複雑系プログラミング特論 個体群生態学でpythonの基礎 (1)

概要

最初のテーマでは、個体群生態学を取り上げます。このクラスを受講している人の多くはあまり生物学や生態学に詳しくないかもしれませんが、複雑系科学とも関係の深いよく知られたモデルを記述することを通して、pythonの基本的な事柄を学びたいと思います。

(注：公開に際し、一部省略した図や記述などがあります。)

個体群生態学

個体群生態学とは、ある生物の群れのダイナミクスの理解を対象とする分野です。個体群、もしくは集団とは、野外の生物集団で、その中で成長や繁殖が行われるまとまりを持ったもののことです。個体数や分布など、様々な環境条件で集団の特性がどのように変移するか、実データと数理解析の両面で理解が進められています。

今回は1種類の個体群の動態に関する簡単な数理モデルを題材に、pythonを使ったモデリングに必要な基礎知識を紹介します。

ロジスティック成長

まず、一番単純な個体群の成長過程を考えます。

”正月にネズミのつがいが12匹の子を生み、2月には親子とも12匹の子を生み・・・と毎月繰り返すと、12月には276億もの多数になる”という計算はねずみ算と呼ばれています。

生物が一定の速度で増殖を続けると、わずかな個体数でもあっという間に膨大な数にふくれあがる話はよく聞かれ、生物の潜在的な増殖能力の大きさを示しているといえます。

しかし、実際には、環境の厳しさや資源の制限、密度増加による競合など、様々な要因によって、可能な個体数に限界が存在します。

このような状況を考慮した集団の成長過程を表現する基礎的なモデルに、ロジスティック成長モデルがあります。

離散ロジスティック成長

ある時刻 t における集団の生物の個体数を N_t としたとき、次の時刻の個体数 N_{t+1} を次の様に書きます。

$$N_{t+1} = N_t + rN_t \left(1 - \frac{N_t}{K}\right)$$

$$N_{t+1} = N_t + rN_t \left(1 - \frac{N_t}{K}\right)$$

パラメータである r は増加の度合いを表す係数で、人口の指数的増加の側面をはじめて強調したマルサスにちなんでマルサス係数、内的自然増加率などと呼ばれています。この値は、その時刻から次の時刻までの出産や死亡流入や流出などの種々のイベントを考慮して計算されます。

もう一つのパラメータである K は環境収容力と呼ばれ、その環境において維持できる最大の個体数を示しています。

pythonでつくってみる

このような個体群の成長過程をpythonでモデリングし、グラフで表示することを考えます。

まず、アルゴリズムを言葉で書いた擬似コードを書いてみます。

- 時刻 t における個体数 N_t に初期値 (1.0) を代入
- 係数 r に値 (1.5) を代入
- 係数 K に値 (100) を代入
- 実験する最大の時刻 T に値 (30) を代入
- 初期の時刻と N_t を表示
- 以下を T 回実行
 - N_t に $N_t + rN_t(1.0 - N_t/K)$ を代入
 - 時刻と N_t を表示

これをpythonで実装すると例えば次のとおりになります。

```
1 Nt=1.0
2 r=1.5
3 K=100
4 T=30
5 print("0"+"t"+str(Nt))
6 for i in range(T):
7     Nt= Nt+r*Nt*(1.0-Nt/K)
8     print(str(i+1)+"t"+str(Nt))
```

IDLE上でのソースコードの作成と編集, 実行

上のソースコードをファイルに記述します。IDLEのFileメニューから、New Windowを選択すると新しいテキスト編集窓が開き、ソースコードを記述できます。コードを実行するにはRunメニューからRun moduleを実行します。ファイルの読み出しと保存はすぐわかるでしょう。

実行すると次のような数列が出力されるはずです。

```
Python Shell
>>>
0      1.0
1      2.485
2      6.119871625
3      14.7378866319
4      33.5866370441
5      67.0456597913
6      100.187342025
7      99.9058025317
8      100.046965637
9      99.9764840951
10     100.011749657
11     99.9941231004
12     100.002937932
13     99.9985309047
14     100.000734515
15     99.9996327343
16     100.000183631
17     99.9999081841
18     100.000045908
19     99.999977046
20     100.000011477
21     99.9999942615
22     100.000002869
23     99.9999985654
24     100.000000717
25     99.9999996413
26     100.000000179
27     99.9999999103
28     100.000000045
29     99.9999999776
30     100.000000011
```

ここでちょっと[pythonの基礎知識](#)へ

可視化

pythonにはmatplotlibというとても強力なグラフ作成パッケージがそろっています。今使っているEPD Freeという環境には、matplotlibを利用するために必要なその他のパッケージなどが全てインストールされています。

matplotlibを使ったグラフの書き方の詳細は別の日に改めて時間を割く予定なので、ここではごく簡単に上のような数列をグラフ化する方法を示します。

まず、matplotlibのパッケージを利用するために、ソースコードの冒頭で以下のように宣言します。

```
1 | import matplotlib.pyplot as plt<br>
```

これで、matplotlibの中のpyplotという関数群を、plt.関数名という形で呼び出せます。

グラフを表示するには、時系列のデータ、この場合は複数のx, y座標のデータを保存する必要があります。ここでは、pythonでよく使われるコンテナであるリストを使います。各時刻（x座標）と個体数（y座標）を順にいれる空のリストを参照するdataX, dataYを他の変数と同様に定義します（リストについては[pythonの基礎知識](#)へ）。

```
1 dataX=[]
2 dataY=[]
```

2つのprint文を次のように置き換えます。

最初の文ではx座標用リストに0を追加（後付け）し、y軸用リストにNtを追加しています。つまり、初期の状態の時刻は0、個体数はNt=1.0ということです。

```
1 dataX.append(0)
2 dataY.append(Nt)
```

forループの中では、iは0からT-1までの間順に増加するので、以下の様書き換えて、順に時間を増しながらそのときの個体数を保存するという作業を行います。

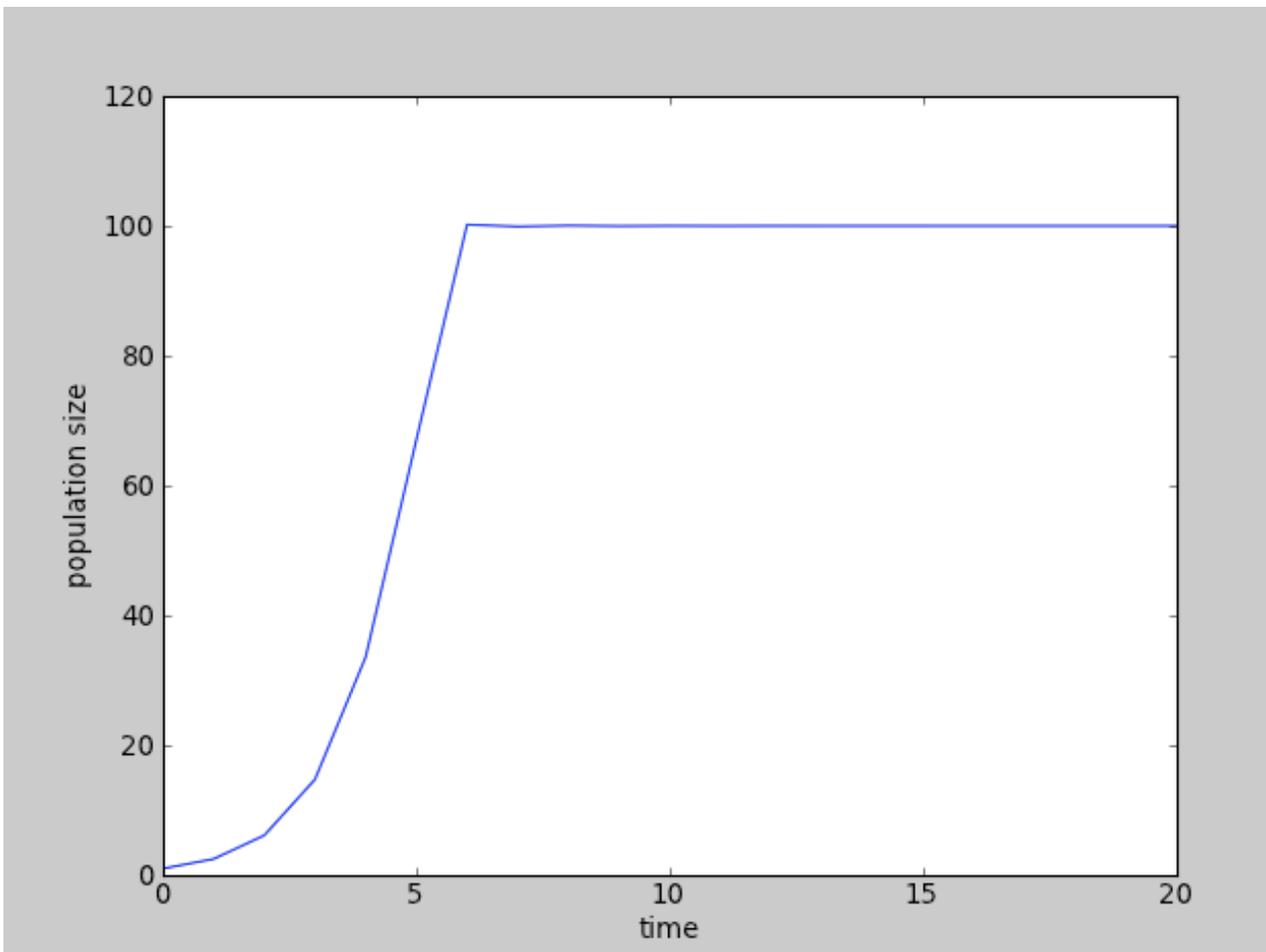
```
1 dataX.append(i+1)
2 dataY.append(Nt)
```

forループの後に以下を付け加えて、結果をグラフ表示します。

```
1 plt.plot(dataX, dataY)
2 plt.xlabel("time")
3 plt.ylabel("population size")
4 plt.show()
```

plt.plot関数は、リストdataXの各値をx座標、dataYの各値をy座標とした点を結ぶ折れ線グラフを描くものです。

実際に書いてみると以下の様になります。



個体群ダイナミクスの解析

上記の様な離散のロジスティック成長モデルは r の値に応じて様々に挙動が変化することが知られています。

例えば、(a) ほぼ単調に増加した後に収束、(b) 一旦急激に増加した後に一定値付近を振動しながら収束、(c) 一定値を中心に不規則な振動を示したりします。

このような挙動を、ロジスティック成長モデルのパラメータ設定を調整することで再現することができます。

関数の定義

様々な r の値による挙動の影響を一つのプログラムで調べるために、各実験パラメータの値を引数として各時刻とその個体数が入ったリストをそれぞれ返す関数を定義します。

関数の定義ではpythonではdefを使います ([pythonの基礎知識へ](#))。pythonでの関数定義はC言語などと概ね同じですが、引数はこれまでと同様型宣言しないで変数名のみ列挙します。

また、関数の返り値についても、冒頭では宣言せずに、直接return(返り値)のように任意の値を指定します。ただし、ここでは関数の返値として、x座標の入ったリストとy座標の入ったリストの

組のように、二つの値を同時に返す、ややpythonに特有な記述法を用いてみます。

例えば、パラメータ K, r, T に加え、 N_t の初期値 N_0 を引数にとって、 x 座標の入ったリストと y 座標の入ったリストの組を返す関数`do_logistic_growth`を考えます。なお、ここでは、引数が関数内でのみ使われるローカル変数と、これまで使ってきたグローバル変数とを区別するために、ローカル変数は小文字で極力表すことにします。

関数を作成し利用するソースコード全文を以下に示します。

```
1  import matplotlib.pyplot as plt
2
3  def do_logistic_growth(k, r, n0, t):
4      nt= n0
5      datax=[]
6      datay=[]
7
8      datax.append(0)
9      datay.append(nt)
10
11     for i in range(t):
12         nt= nt+r*nt*(1.0-nt/k)
13         datax.append(i+1)
14         datay.append(nt)
15
16     return(datax, datay)
17
18
19 dataX, dataY= do_logistic_growth(100.0, 1.5, 1.0, 20)
20 plt.plot(dataX, dataY)
21
22 plt.xlabel("time")
23 plt.ylabel("population size")
24 plt.show()
```

ここでの注意点は二つ。一つは、関数定義は関数を利用する前に書くということです。

もう一つは、`return`文の引数の記述`return(fdataX, fdataY)`と、その返り値を割り当てる`dataX, dataY=`という記述です。このように、`return`文の引数に複数の値を並べて書く事ができます。これを受けるときは、同数の変数を順に対応させてカンマ区切りで列挙します。

この場合、`dataX`には関数内のリスト`datax`が、`dataY`には関数内の`datay`が割り当てられます。なので、その下で、`dataX, dataY`をそのまま`plot`関数の引数にして、グラフを描くことができます。

一つの関数から複数の値を受け渡す方法はいろいろあるけれども、複数の割り当てを並列に一文で行えるのはpythonの便利なところの一つ。使い次第ではコーディングが楽になったり、見やすくなったりします。

練習 1

上で挙げた(a), (b), (c)それぞれの値の推移に類似した挙動 (a:単調増加->収束, b:振動->減衰, c:不規則) を示す r の値 (但し $0 < r \leq 3$ の範囲で) を探し, 時刻20までの挙動を示した結果を同時に一つのグラフで表しなさい. なお, $K=100, N_0$ (個体数 N_t の初期値) $=1.0$ とする. その際の各条件での r の値も同時に示しなさい.

ただし, 以下のヒントを参考にすること.

複数線の同時プロット

`show`関数を実行する前に, `plot`関数を複数回実行することで, 一つのグラフに複数の線をプロットできます.

つまり, 引数として指定するパラメータ値をかえながら,

```
1 dataX, dataY= do_logistic_growth(パラメータ設定1)
2 plt.plot(dataX, dataY)
3
4 dataX, dataY= do_logistic_growth(パラメータ設定2)
5 plt.plot(dataX, dataY)
6
7 ...
```

などと繰り返すことで, 異なる設定での推移を重ねて表示できます.

分岐図を描く

あるパラメータ設定を横軸にとり, 縦軸にそのパラメータ設定における系の挙動, 具体的には, 系が収束する先を表した図を分岐図と呼びます.



つまり, 注目するパラメータ条件のどの付近で, 系の挙動が別れるかがこの図を見ることではっきりわかります.

練習2

そこで、横軸をロジスティック成長モデルにおける r 、縦軸をその r の設定で実験したときの時刻 251 から 300（最後の 50 ステップ）における N_t の値とした分岐図を作成してみましょう。但し、 $K=100$, r は 1.0 以上 3.0 以下の範囲で 0.01 刻みとします。

以下のヒントを頼りに作成してみましょう。

ヒント 1) 分岐図を描くには線ではなくて点をプロットするのが一般的。点でプロットするには、以下のように、`plt.plot`関数の引数にもう一つ、「`.'`」を加える。

```
plt.plot(dataX, dataY, '.')
```

ヒント 2) 次のようなデータ保存用リスト `bifX`, `bifY` を点でプロットすればよい。

```
bifX: [1.0, 1.0, 1.0, ..., 1.0 (50個),
```

```
       1.01, 1.01, ..., 1.01 (50個),
```

```
       ...,
```

```
       3.0, ..., 3.0 (50個)]
```

```
bifY: ["r=1.0, 時刻251の時のNt", "r=1.0, 時刻252の時のNt", ..., "r=1.0, 時刻300の時のNt",
```

```
       "r=1.01, 時刻251の時のNt", "r=1.01, 時刻252の時のNt", ..., "r=1.01, 時刻300の時のNt",
```

```
       ...,
```

```
       "r=3.0, 時刻251の時のNt", "r=3.0, 時刻252の時のNt", ..., "r=3.0, 時刻300の時のNt"]
```

つまり、はじめに、上のデータを入れる空のリスト `bifX`, `bifY` を作成し、次に、各 r の設定で実験した結果のリスト（ただし `dataX[251:301]` と `dataY[251:301]`）を、`bifX`, `bifY` に次々結合して大きくしていく繰り返し操作を行い、最後にそれをデータとしてグラフ表示すればよい。

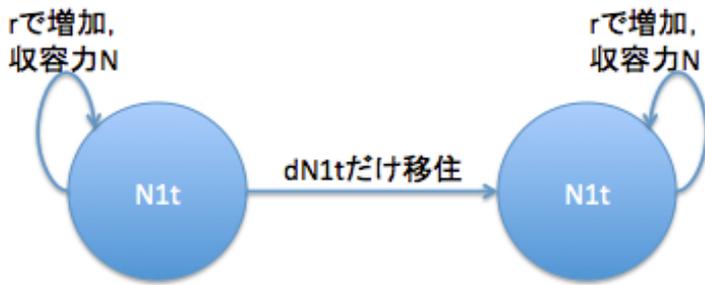
(リストの扱いに関するヒント。その他は文法のページを参照のこと。)

- リスト `x` 中の a 番目から b 番目までの要素からなる部分リストを表すには、`x[a:b+1]` とする。
- リスト `x` と `y` を結合するには、`x+y` と書く。 `x=[x1, x2]`, `y=[y1, y2]` なら、`x+y` は `[x1, x2, y1, y2]`
- a 個の値 b からなるリスト `x` は、`x=[b]*a` と書ける。この場合、`x` の中身は `[b, b, ..., b]` で、中の b の数は a 個。

■ 移住のある環境

2つのパッチ（生物集団が存在可能な場所）において、ある種がそれぞれロジスティック成長している状況を考えます。パッチ1内における時刻 t での注目種の個体数を $N1t$ 、パッチ2での注目種の個体数を $N2t$ とします。簡単化のため、パッチ1, 2の各集団の内的自然増加率を共に r とし、注目する種の環境収容力もどちらも同じく K とします。

ここで、パッチ1からパッチ2への個体の移住がある状況を考えます。



具体的には、次ステップでは、現ステップでの個体数に比例した個体数だけ、パッチ1からパッチ2へ個体が移動するものとします。このときの比例係数、つまり移住率を $d(0 \leq d \leq 1)$ とします。

練習3

- 1) 各集団のロジスティック成長の一般式に、上記の移住の影響を表す新たな項を追加して、この状況を表したモデルを作成し、その式を示しましょう。ただしこの項に対する環境収容力の影響は考慮しなくてかまいません。
- 2) このモデルにおいて、移住率 d は両集団の挙動にどのような影響を与えうるでしょうか。プログラムを書いて計算機実験を行い、その傾向がよくわかる図を描いて説明しましょう。（なお、簡単化のため、グラフを作成する場合は $K=100$ 、各集団の初期個体数1とします。）

迅速な進化と個体群動態

生物には、これまで注目してきた個体数の変化などの生態的な変化以外にも、進化、つまり、集団における遺伝子構成の変化も存在します。

従来、前者は比較的短い時間スケールで生じる一方、後者はより長い時間スケール（何万世代とか）で生じると考えられてきました。しかし、最近の研究では、進化も数世代や数十世代といった生態的な変化に近い早い時間スケールで生じ、個体群のダイナミクスに影響する場合があります。明らかになってきています。

被食者の進化を導入した被食・捕食関係

例えば、被食者の遺伝的特性が多様であり、捕食者に対する耐性（コストを払って捕食を逃れること）が進化する場合、上記の典型的なロトカボルテラ系の被食・捕食の個体数変動とは異なる挙動が生じることが知られています。

例えば、以下の簡略化したモデルを考えます。

被食者には異なる遺伝子を持つ2種AとBが存在し、それぞれの個体数を $N_{A,t}$ と $N_{B,t}$ として、以下の通りにロトカボルテラ系を改変します。

$$NA_{t+1} = NA_t + \alpha NA_t (r_A - X - d_A P_t)$$

$$NB_{t+1} = NB_t + \alpha NB_t (r_B - X - d_B P_t)$$

$$P_{t+1} = P_t + \alpha P_t (Q - s)$$

$$X = NA_t + NB_t$$

$$Q = d_A NA_t + d_B NB_t$$

r_A , r_B はそれぞれ被食者A, Bの内的自然増加率, d_A , d_B はそれぞれ被食者A, Bの捕食者に対する脆弱性（見つけやすさや捕食されやすさ）を表す係数です。また, X は被食者の総数, Q は捕食者が被食者を発見する確率にそれぞれ対応します。

X を含む項は被食者の数が増えると被食者同士で競合が起きて増加率が低下すること, d を含む項は捕食者に見つかった（防御できなかった）分だけ食べられることを表しています。また, 捕食者の式の Q は被食者を見つけられた分だけ食べて増えることを示しています。

（注：上記の式は”Yoshida T, Ellner SP, Jones LE, Bohannan BJM, Lenski RE, et al. (2007) Cryptic population dynamics: Rapid evolution masks trophic interactions. PLoS Biol 5(9): e235.”を元に作成）

練習 4

ここで, 被食種Bは被食種Aよりも脆弱性が低い一方, 耐性のためのコストにより内的自然増加率も低い状況を考えます。つまり, $d_B < d_A$ かつ $r_B < r_A$ とします。

このとき, 各パラメータを適宜設定して実験を行い, 捕食者Pの個体数 (P), 被食者A (NA), B (NB) それぞれの個体数と被食者の個体数の総和 ($NA+NB$) の推移を示すグラフを描きましょう。特に, 従来のロトカボルテラ系の挙動と大きく異なると解釈できる状況を探して示しましょう。また, なぜその様なことが生じるか, 考察してみましょう。

ヒント：“迅速な進化”をキーワードにして調べてみよう。