

Modelação da Evolução de Espécies

1 Objetivos

Com este projeto pretende-se que os alunos desenvolvam uma aplicação em linguagem *Java* (Horstmann, 2015) onde apliquem um processo básico de desenvolvimento de aplicações informáticas, valorizando todas as fases do ciclo de desenvolvimento, desde a análise e conceção aos testes de validação. Pretende-se também que os alunos elaborem um relatório que descreva a aplicação concebida, o processo de desenvolvimento e que apresentem e critiquem os resultados obtidos.

Em particular, no projeto a realizar no corrente ano letivo pretende-se que os alunos estudem um modelo para prever a evolução de espécies e desenvolvam uma aplicação que implemente esse modelo e permita a qualquer biólogo analisar a evolução de uma determinada espécie.

2 Plano de Trabalho

Para desenvolver o projeto de LAPR1 o aluno terá que, numa primeira fase, compreender os objetivos do projeto e estudar a modelação da evolução de espécies, mais precisamente, terá que estudar o modelo introduzido por Lotka em 1920 e formalizado por Leslie (Leslie, 1945). Sendo que a formalização deste modelo recorre a ferramentas de álgebra linear, em especial valores e vetores próprios (Larson, 2012), o aluno também terá que realizar um estudo aprofundado destas ferramentas.

Após adquirir os conhecimentos necessários o aluno terá que implementar uma aplicação que permita a qualquer biólogo estudar a evolução de uma determinada espécie utilizando o modelo de Lotka-Leslie. Esta aplicação deve incluir um conjunto de módulos que permitam: ler e escrever dados em ficheiros; modelar a evolução das espécies utilizando o estado inicial (representado por um vetor) de uma população e um modelo de evolução (representado por uma matriz); calcular estatísticas simples que caracterizem a evolução da população ao longo do tempo. Todos os métodos desenvolvidos devem ser testados recorrendo a testes unitários.

3 Modelação da Evolução de um Espécie

Ao longo dos anos foram propostos vários modelos matemáticos com o objectivo de prever a evolução de determinadas espécies. O modelo de população mais simples, introduzido por Lotka em 1920 e formalizado por Leslie (Leslie, 1945) vinte anos mais tarde, é baseado nas taxas de mortalidade e fecundidade para diferentes intervalos de idade.

Considerando que x_i^t é o número de indivíduos reprodutores (tipicamente, do sexo feminino) cujas idades no tempo t pertencem ao i -ésimo intervalo ($i = 0, \dots, n$). Seja x_i^0 o número de indivíduos reprodutores do intervalo de idade i no momento inicial ($t = 0$), que é conhecido, s_i a taxa de sobrevivência dos indivíduos reprodutores que pertencem ao i -ésimo intervalo, e F_i o número médio de indivíduos reprodutores gerados por um indivíduo reprodutor no i -ésimo intervalo de idade. O modelo de Lotka-Leslie é definido pelas equações:

$$x_{i+1}^{t+1} = x_i^t s_i, \quad i = 0, \dots, n-1$$

$$x_0^{t+1} = \sum_{i=0}^n x_i^t F_i,$$

em que as primeiras n equações descrevem o desenvolvimento da população e a última equação representa a sua reprodução.

Em notação matricial, podemos escrever

$$\mathbf{x}^{t+1} = A\mathbf{x}^t,$$

em que $\mathbf{x}^t = (x_0^t, \dots, x_n^t)^T$ e A é a matriz de Leslie:

$$A = \begin{pmatrix} F_0 & F_1 & \cdots & \cdots & F_n \\ s_0 & 0 & \cdots & \cdots & 0 \\ 0 & s_1 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & s_{n-1} & 0 \end{pmatrix}$$

Considerando que no momento inicial a população é descrita pelo vetor \mathbf{x}^0 , também é possível prever a distribuição da população (na geração k) calculando potências da matriz de Leslie:

$$\mathbf{x}^t = A^t \mathbf{x}^0.$$

Pode mostrar-se que a dinâmica desta população é determinada pelo valor próprio de módulo máximo de A , digamos λ_1 , enquanto que a distribuição dos indivíduos nos diferentes intervalos de idade (normalizada pela população total), obtém-se como o limite de \mathbf{x}^t para $t \rightarrow +\infty$ e verifica $A\mathbf{x} = \lambda_1\mathbf{x}$.

Este tipo de modelo é muito utilizado para estimar a evolução de uma determinada população (e.g., estudar a evolução demográfica de um determinado país), permitindo prever a dimensão da população num determinado momento t ($N_t = \sum_{i=0}^n x_i^t$) e a taxa de variação ao longo dos anos ($\Delta_t = \frac{N_{t+1}}{N_t}$).

4 Trabalho a Desenvolver

O trabalho a realizar até ao dia 24 de Janeiro de 2021 consiste no:

- Estudo do modelo de população formalizado por Leslie.
- Estudo do cálculo de valores e vetores próprios.
- Desenvolver uma aplicação informática que permita estudar a evolução de uma determinada população. Esta aplicação deve implementar o modelo de população formalizado por Leslie. O cálculo dos valores e vetores próprios deve ser realizado com o auxílio da biblioteca `la4j` - linear algebra for Java (<http://la4j.org/apidocs/>).
- A aplicação deverá permitir introduzir os coeficientes não nulos da matriz de Leslie (entradas s_i e F_i , ver Secção 3) e um vetor a descrever a distribuição inicial da população de duas formas:
 - Carregando um ficheiro de texto (txt) que contém o vetor inicial e a matriz;
 - Utilizando um interface, simples e intuitivo, que permita ao utilizador introduzir cada um dos coeficientes da matriz.

- A aplicação deve incluir uma funcionalidade que permita ao utilizador introduzir o número de gerações a estimar. Após identificar o número de gerações, a aplicação deve gerar a distribuição da população (e também a distribuição normalizada pelo total da população) para cada instante de tempo.
- A aplicação também deve incluir funcionalidades que permitam calcular a dimensão da população num determinado momento (N_t) e a taxa de variação ao longo dos anos Δ_t .
- Implementar uma funcionalidade que permita analisar o comportamento assintótico da população associado ao maior valor próprio.
- Desenvolvimento de módulo que permita gerar a representação gráfica da dimensão da população (total de indivíduos), a taxa de variação e a evolução da distribuição da população (e também a distribuição normalizada pelo total da população), por classe, ao longo do tempo. Este módulo deverá permitir gerar um de três formatos gráficos distintos: png, txt ou eps.

Para implementar este módulo, os alunos terão obrigatoriamente que recorrer à aplicação *gnuplot* (Janert, 2010; Williams & Kelley, 2010), à classe *java.lang.Runtime* e às classes utilizadas na unidade curricular Algoritmia e Programação (APROG). Os alunos não poderão utilizar bibliotecas que implementam interfaces para a biblioteca *gnuplot*, tais como as bibliotecas *jgnuplot* e *JavaPlot*.

- A aplicação deve incluir uma funcionalidade que permita executar todas as funcionalidade através da execução de um único comando. Neste comando serão especificados todos os parâmetros, o ficheiro de entrada e o ficheiro de saída.
- Elaborar um relatório em que: é feita uma breve introdução ao modelo de Lotka-Leslie, à aplicação dos valores e vetores próprios; é descrita a metodologia de trabalho que utilizaram para desenvolver a aplicação; é descrita a implementação da aplicação; e é feita uma análise dos resultados. A descrição da implementação da aplicação deve incluir um diagrama que identifique claramente os módulos e suas dependências. A apresentação das medidas e valores e vetores próprios deve incluir exemplos ilustrativos.
- No relatório também devem explorar dois casos de estudo:
 - Estudar o crescimento de pombos na cidade costeira MarSol (nome fictício), supondo que o número inicial de pombos é igual a 1000 juvenis, 300 jovens, 330 adultos e 100 maduros e que as taxas de fertilidades e sobrevivência para os juvenis, jovens, adultos e maduros são os definidos na seguinte matriz:

$$\begin{pmatrix} 0 & 3.00 & 3.17 & 0.39 \\ 0.11 & 0 & 0 & 0 \\ 0 & 0.29 & 0 & 0 \\ 0 & 0 & 0.33 & 0 \end{pmatrix}$$

- Estudar o crescimento de gaivotas na cidade costeira MarSol (nome fictício), supondo que o número inicial de gaivotas é igual a 600 juvenis, 200 jovens, 130 adultos e 40 maduros e

que as taxas de fertilidade e sobrevivência para os juvenis, jovens, adultos e maduros são os definidos na seguinte matriz:

$$\begin{pmatrix} 0 & 3.5 & 1.5 & 0.39 \\ 0.4 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{pmatrix}$$

- Todos os resultados apresentados devem ser obtidos recorrendo à aplicação desenvolvida pelo grupo.

4.1 Formato dos Ficheiros de Entrada e Saída

Os dados de entrada para a aplicação são as entradas não nulas da matriz de Leslie (entradas s_i e F_i , ver Secção 3) e um vetor a descrever a distribuição inicial da população. Todos os campos dos ficheiros estão separados por uma vírgula. O nome do ficheiro identifica a espécie/população.

Um exemplo do conteúdo de um ficheiro de entrada contendo a descrição do vetor inicial e das entradas não nulas da matriz Leslie:

x00=20, x01=10, x02=40, x03=30

s0=0.50, s1=0.80, s2=0.50

f0=0.50, f1=2.40, f2=1.00, f3=0.00

Relativamente aos ficheiros de saída, teremos dois tipos de ficheiros. Um tipo de ficheiro são os ficheiros que representam gráficos de evolução de uma determinada espécie. Sempre que visualiza um gráfico, o utilizador deve ter a opção de gravar. O outro tipo de ficheiro é um ficheiro de texto (txt) que resulta de executar a aplicação em modo não interativo (ver Secção 6). Cada um dos ficheiros de saída deve ter um nome que permita identificar a operação realizada, o nome da população e a data em que o ficheiro foi gerado. Os grupos que não consigam gerar gráficos utilizando *gnuplot* devem gerar ficheiros de texto (txt) que representem as populações ao longo do tempo.

5 Método de Trabalho

- Todos os alunos devem utilizar a metodologia de trabalho definida no *eduScrum* (Delhij & Solingen, 2013). Cada um dos grupos deve escolher dois *Scrum Masters*, que serão responsáveis por gerir a execução de tarefas. O primeiro *Scrum Master* é responsável pela gestão de tarefas desde o início do projeto e até ao dia 13 de Janeiro às 24h00m, enquanto o segundo *Scrum Master* é responsável por gerir as tarefas entre o dia 14 de Janeiro às 00h00m e a data de submissão final do projeto. Para atingir os objetivos, o grupo deve utilizar a ferramenta *Trello* e registar as tarefas do projeto, a atribuição de tarefas, o estado de cada tarefa e as tarefas concluídas.
- A aplicação será desenvolvida utilizando o sistema de controle de versões *Git* e o *Bitbucket* (<https://bitbucket.org>). Todos os alunos terão que criar uma conta no *Bitbucket* com o endereço

de email do ISEP (i.e. XXXXXXX@isep.ipp.pt) e cada grupo terá que criar um repositório. A designação do repositório deve seguir o formato: "LAPR1_TurmaDAB_Grupo01". O repositório deve ser partilhado com todos os docentes que lecionam a turma onde o grupo está inserido.

- O grupo deve criar uma pasta no OneDrive onde guarda todo o material desenvolvido para a realização do projeto. A designação da pasta deve seguir o formato do exemplo "LAPR1_TurmaDAB_Grupo01". A pasta será partilhada com todos os docentes que lecionam a turma onde o grupo está inserido. Não é necessário incluir nesta pasta o código que está disponível no repositório do *BitBucket*.

6 Processo de Desenvolvimento de Software

- A aplicação deve ser estruturada e organizada em módulos. Será valorizada uma correta decomposição modular e o reaproveitamento de módulos.
- O trabalho deverá ser desenvolvido em linguagem *Java* e deverá resultar num ÚNICO projeto.
- Para o cálculo dos valores e vetores próprios será utilizada a biblioteca *la4j* - linear algebra for Java (<http://la4j.org>). Esta biblioteca não pode ser utilizada em outras operações que não seja o cálculo de valores e vetores próprios.
- A aplicação pode ser executada em modo interativo ou sem interação por parte do utilizador.
 - No modo interativo, a aplicação deverá ser chamada da linha de comandos utilizando o comando: `java -jar nome_programa.jar` ou `java -jar nome_programa.jar -n nome_ficheiro_entrada.txt`, no caso em que os dados são carregados utilizando um ficheiro. Neste modo todos os parâmetros necessários são solicitados em tempo de execução ao utilizador.
 - No modo não interativo, o utilizador tem que identificar todo os parâmetros na linha de comando por forma a identificar o ficheiro de entrada, o número de gerações (momentos), se pretende calcular o valor e o vetor próprio, a dimensão da população a cada geração, a variação da população entre gerações, e o nome do ficheiro de saída. Neste modo o comando terá a seguinte sintaxe:
`java -jar nome_programa.jar -t XXX -g Y -e -v -r nome_ficheiro_entrada.txt nome_ficheiro_saida.txt`. Neste caso, o valor associado ao `t` (XXX) identifica o número de gerações, o valor associado ao `g` (Y) identifica o formatos dos ficheiro dos gráficos gerados pelo gnuplot (Y pode tomar os valores 1, 2 e 3, em que o valor 1 representa o formato png, 2 o formato txt e 3 o formato eps) e os parametros `e`, `v`, `r`, quando especificados, identificam a obrigatoriedade de calcular o valor e o vetor próprio (`e`), a dimensão da população a cada geração (`v`), a variação da população entre gerações (`r`).
- Todos os métodos desenvolvidos terão, obrigatoriamente, de estar associados a testes unitários. Por exemplo, se o aluno criar o método `find_max_eigenvalue(matrix)` para determinar o maior valor próprio de uma matriz de comparação, também deve criar o método `test_find_max_eigenvalue(matrix, expectedMaxEigenValue)` (ver Algoritmo 1), em que `expectedMaxEigenValue` é o valor do maior valor próprio conhecido da matriz. Estes testes são extremamente úteis para determinar se os métodos estão de acordo com a sua especificação e se a edição destes não alterou a funcionalidade.

```
Bool test_find_max_eigenvalue(matrix, expectedMaxEigenValue)
{

    maxEigenvalue = find_max_eigenvalue(matrix);

    if(expectedMaxEigenValue==maxEigenvalue)
        return True;
    else
        return False;

}
```

Algoritmo 1: Exemplo de métodos de teste unitários

7 Submissão do Trabalho

Datas e entregas de trabalho a efetuar através do Moodle:

- Dia 24 de Janeiro de 2021, até às 23h00m
 - Submeter o projeto desenvolvido, versão final, incluindo toda a estrutura de diretorias e ficheiros do projeto (incluindo o executável), num único ficheiro comprimido (ZIP).
 - Relatório em formato pdf não ultrapassando as 25 páginas. A escrita do relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de competências).

Nota: Os ficheiros deverão identificar, obrigatoriamente, a designação do grupo e a turma a que os alunos pertencem (Exemplo: "LAPR1_TurmaDAB_Grupo01_projeto.ZIP" e "LAPR1_TurmaDAB_Grupo01_relatorio.PDF").

Referências

- Delhij, A., & Solingen, R. (2013). *The eduscrum guide: The rules of the game*. (Disponível em http://eduscrum.nl/file/CKFiles/The_eduScrum_Guide_EN_December_2013_1.0.pdf)
- Horstmann, C. (2015). *Big java: Early objects, 6th edition*. Wiley. Retrieved from <https://books.google.pt/books?id=ib12CwAAQBAJ>
- Janert, P. K. (2010). *Gnuplot in action : understanding data with graphs*. Manning Publications. Retrieved from <http://opac.inria.fr/record=b1133156>
- Larson, R. (2012). *Elementary linear algebra*. Cengage Learning.
- Leslie, P. H. (1945). On the use of matrices in certain population mathematics. *Biometrika*, 33(33), 183–212.
- Williams, T., & Kelley, C. (2010, March). *gnuplot 5.0: An interactive plotting program*. <http://gnuplot.sourceforge.net/>.