PROJECTO DE ALGORITMOS E MODELAÇÃO COMPUTACIONAL

AGRUPAMENTO (CLUSTERING) PARA MODELOS FARMACOCINÉTICOS

Professor: Paulo Mateus
Aluno: Ricardo J. D. Ferreira
Mestrado Integrado em engenharia Biomédica (69407)
Instituto Superior Técnico
3 de Junho de 2014

INTRODUÇÃO

O projecto consiste em descobrir um conjunto de parâmetros θ_j relativos a *j* gaussianas, de segundo um algortimo de *Expectation–maximization*. Este algoritmo, mas adaptado a uma *Gaussian Mixture Model*, permite encontrar os parâmetros das gaussianas que maximizam a verosimilhança dos dados.

Tal como descrito no enunciado, é necessário iterar o processo até a uma dada condição de paragem, em que os parametros b_{dj} não variam muito de iterada para iterada, sendo assim os parametros de cada gaussiana os ideiais para aproximar e fazer *clustering* aos dados (sendo este *clustering não-supervisionado*).

Deste modo, a função $f(\theta_j, t) = \sum_{i=1}^{2} a_{ij} e^{-b_{ij}t}$ é a que se quer estimar, para um conjunto de dados com j gaussianas.

IMPLEMENTAÇÃO

DADOS PARA A PRIMEIRA ENTREGA

Para a primeira entrega, foram pedidos pelo professor as implementações de três classes, sejam elas a *"Amostra"*, *"Grafos de compartimentos"* e *"Misturas de gaussianas"*. De seguida serão enunciadas as justificações à forma como se procedeu à escolha de cada tipo de dados para determinada classe.

"AMOSTRA" (AMOSTRA.JAVA)

Nesta classe, optou-se por se organizar os dados da amostra, neste caso o i_d do paciente, um tempo e uma concentração, numa lista ligada, em que cada nó representava um vector contituido por 3 entradas, sendo as características acima descritas.

Sendo relativamente fácil operar sobre listas simplesmente ligadas, a adição (ordenada!) e manipulação de elementos nas mesmas são operações de baixa complexidade.

Essas mesmas listas ligadas estão dissimuladas numa *ArrayList* do java, tornando o processo de manuseamento da mesma muito mais rápido.

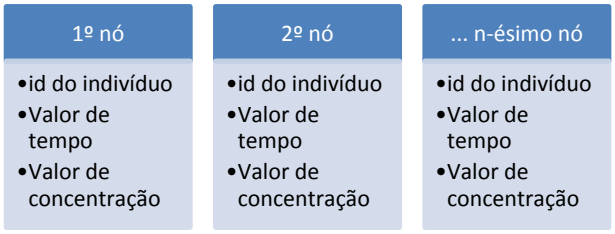


FIGURA 1: ESQUEMA DA IMPLEMENTAÇÃO DOS NÓS LIGADOS NA CLASSE *AMOSTRA*

"GRAFOS DE COMPARTIMENTOS" (GRAFOO.JAVA)

Como o nome da classe indica, foi necessário implementar grafos que simulassem os compartimentos, neste caso, os diferentes orgãos do corpo humano, em que cada nó seria um orgão, e cada aresta a interação entre esses mesmos orgãos.

No presente projecto, é apenas considerado um compartimento, neste caso simular-se-á a entrada e saída de fármacos de todo o organismo humano.

Foi escolhida a implementação de grafos com uma *Matriz de Adjacência*, neste caso um *"Tensor de Adjacência",* já que nas duas primeiras dimensões se indicam os compartimentos de partida e de chegada, e a aresta é representada pela terceira e quarta dimensões, em que um tensor possui a estimativa inicial para a mistura de gaussianas entre esses dois compartimentos.

A título de exemplo, dados n compartimentos, tem-se uma matriz n x n x 4 x 6, como seria de esperar.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix} \end{array}$$

"MISTURAS DE GAUSSIANAS" (MIX.JAVA)

Como indicado no enunciado do projecto, a classe de mistura de gaussianas implementa rotinas que permitem operações sobre o método de misturas de gaussianas, sendo a principal a prob, que devolve a probabilidade de uma lista de pontos ser observado por uma mistura.

Também se pode actualizar os parâmetros de uma mistura após cada ciclo do algoritmo EM.

ALTERAÇÕES À PRIMEIRA ENTREGA

"AMOSTRA" (AMOSTRA.JAVA)

A alteração feita aos dados da primeira entrega feita na classe amostra, em que uma implementação usando os dados primitivos de java (*ArrayList*) em substituição à implementação de listas ligadas anteriormente feita, permite agora ordenar, inserir e procurar dados na amostra quer a partir do início do mesmo, quer a partir do fim, sendo a pesquisa de dados na mesma mais eficiente.

Também foi adicionada uma rotina *amostra.individuos()* que devolve os índices dos indivíduos presentes na amostra.

"GRAFOS DE COMPARTIMENTOS" (GRAFOO.JAVA)

Nesta classe, quase todas as funções foram alteradas, devido a uma incompreensão sobre o enunciado do projecto, o que levou à mudança da definição de aresta do grafo, que anteriormente continha os parâmetros para apenas uma gaussiana, e agora contém os parâmetros relativos a uma mistura.

"MISTURAS DE GAUSSIANAS" (MIX.JAVA)

Nesta classe foi retirado o parâmetro M, referente ao número de gaussianas na mistura, uma vez que o mesmo é definido pelos parametros dados no ficheiro init.

Também algumas atribuições matriciais foram reescritas, de modo a aumentar a eficiência das rotinas nesta classe.

SEGUNDA ENTREGA

"LEITURA DE DADOS" (CSVREADER.JAVA)

Nesta classe, foram implementadas duas rotinas: uma para ler os ficheiros *EM#### .csv,* de modo a ser possível criar a amostra a partir de dados fornecidos no exterior, e outra para ler os ficheiros init.csv, de modo a ler os valores para as estimativas iniciais de θ para uma aresta.

Estas rotinas fazem uso das classes implementadas anteriormente, e transformam os dados raw presentes nos ficheiros fornecidos pelo docente em dados que seguem as estruturas implementadas em *amostra.java, mix.java* e *grafoo.java,* devolvendo, no caso da rotina reader uma amostra, e no caso da rotina readertheta uma mistura.

"INTERFACE GRÁFICA" (GUI.JAVA)

A interface gráfica permite ao utilizador carregar dados relativos à amostra (ou amostras), aos dados iniciais, e introduzir o número de nós do grafo representativo dos vários

"compartimentos", e a respectiva ligação (caso exista).

No final de todo o procesoo permite ao utilizador guardar os dados num ficheiro *.txt*.

Mais adiante, no manual do utilizador, serão discriminados todos os campos da interface gráfica, e como devem ser preenchidos de modo a iniciar o processo e obter os dados resultantes do algortimo.

"ALGORITMO DE APRENDIZAGEM NÃO SUPERVISIONADA - EM" (EM.JAVA)

Esta é talvez a classe mais importante implementada em todo o projecto. Com a mesma, e usando o método *em.alg((args))* é possível obter os parâmetros das gaussianas que melhor descrevem os dados que serviram de input.

A ordem de actualização dos parametros foi a seguinte:

- $X_{ij}^{(k)}$;
- $w_j^{(k+1)}$;
- $b_{1j}^{(k+1)}$;
- $a_j^{(k+1)}$;
- $b_{2j}^{(k+1)}$;
- $\sigma_j^{(k+1)}$;

A implementação por esta ordem, e com $a_j^{(k+1)}$ a depender de $b_{1j}^{(k+1)}$, e consequentemente $b_{2j}^{(k+1)}$ a depender de $b_{1j}^{(k+1)}$ e $a_j^{(k+1)}$ fez com que o algoritmo convergisse mais rapidamente para a solução óptima.

O mesmo pára quando todos os b_{dj} satisfazem a condição de paragem descrita no enunciado deste projecto.

MANUAL DE UTILIZAÇÃO

De seguida serão descritos os passos necessários para visualizar os resultados da aplicação do algoritmo.

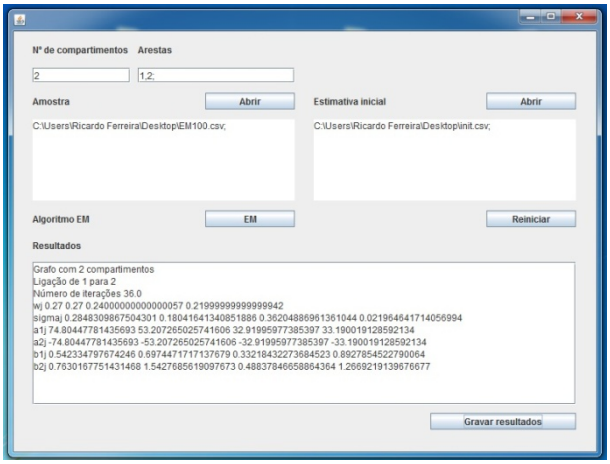


FIGURA 2: JANELA DE INTERFACE GRÁFICA

A janela representada na figura 2 possui alguns campos a preencher, com as seguintes características:

- No campo *"Nº de compartimentos"* o número de nós do grafo, isto é, o número de compartimentos necessário para a simulação;
- No campo *"Arestas"* são colocadas ligações separadas por ponto e virgula, e nó de partida e chegada por virgula. Deste modo, caso se queira colocar uma ligação do nó 1 para o nó 2, deve indicar-se assim

  1,2;

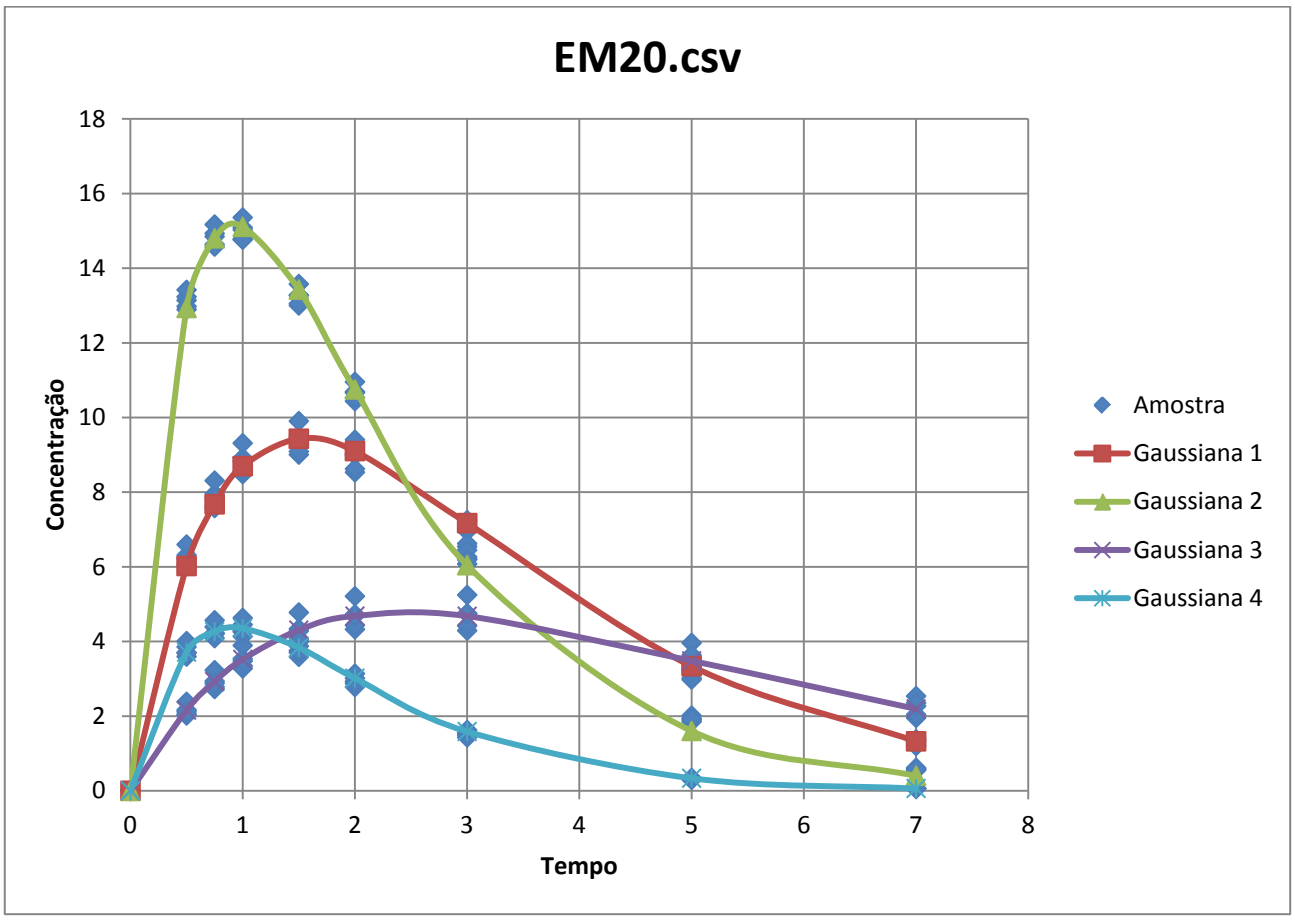
- De seguida devem abrir-se os ficheiros relativos às amostras e condições iniciais para as misturas de gaussianas correspondentes a cada nó, os quais ficarão representados na interface para o utilizador poder confirmar;
- Para correr o algoritmo basta pressionar o botão *"EM"*;
- Caso seja necessário corrigir os valores introduzidos na interface o botão reiniciar deve ser pressionado;
- No final de todo o processo, e quando todos as arestas do grafo são processadas, o resultado é indicado na janela, com o botão *"Gravar resultados"* a permitir guardar os mesmos num ficheiro *.txt*.

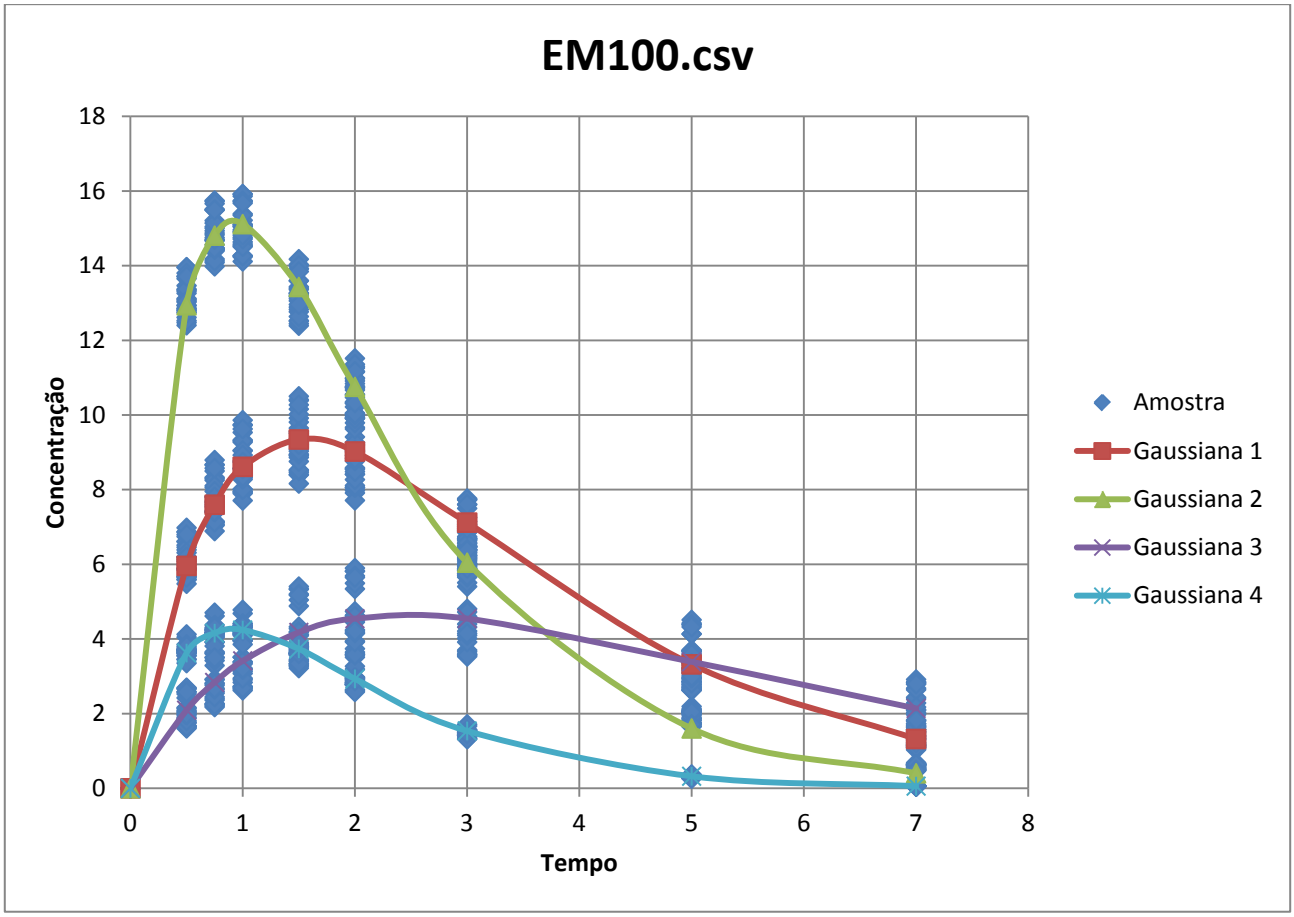
RESULTADOS

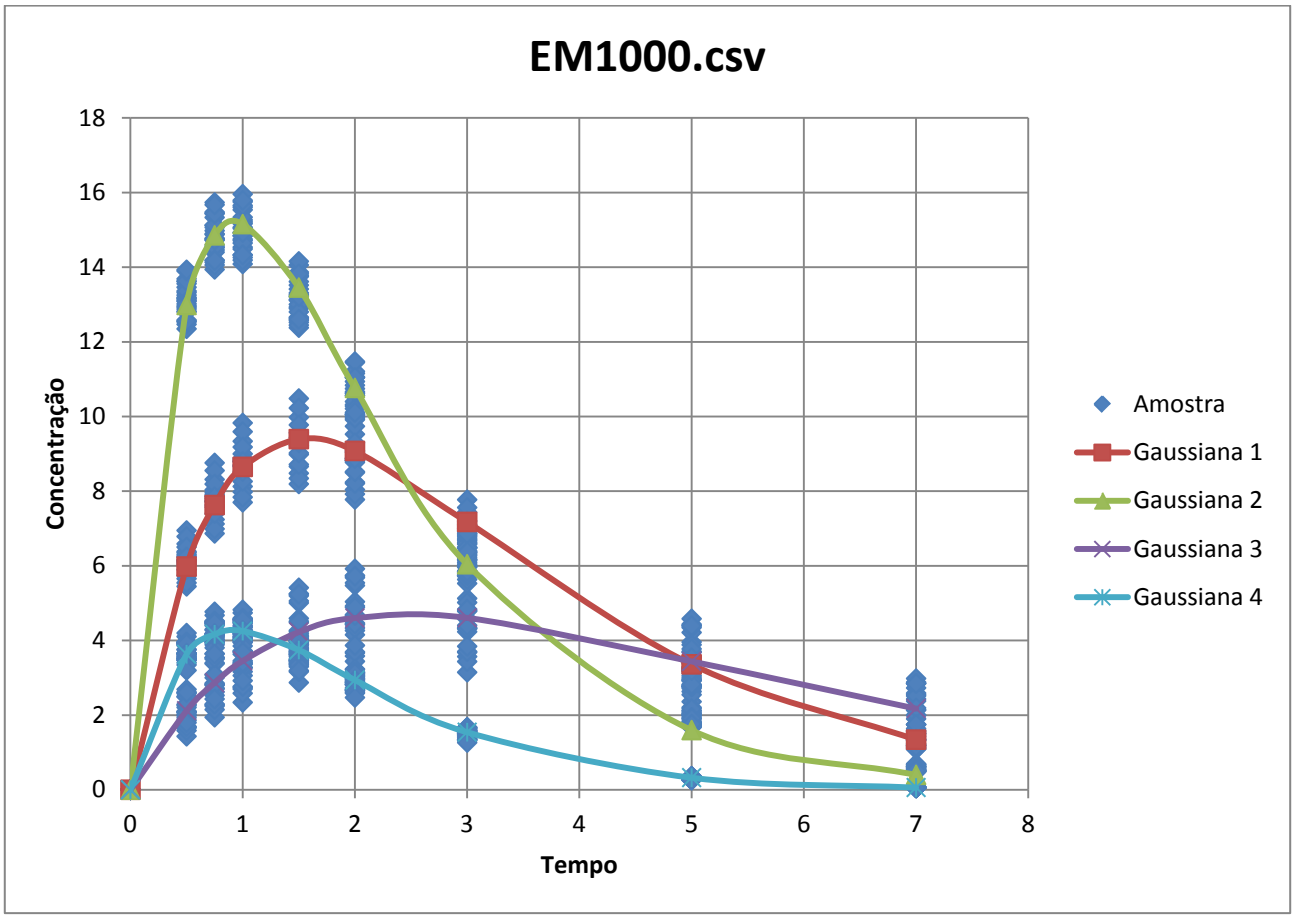
De notar que os resultados são impressos em ficheiros *.txt*, e os mesmos contêm toda a informação necessária sobre a Mistura de Gaussianas. Noutros ficheiros anexos a este relatório estão alguns exemplos de resultados para as amostras *EM20.csv, EM100.csv e EM1000.csv*, sejam eles *resultadosEM20.txt, resultadosEM100.txt* e *resultadosEM1000.txt.*

Também serão anexados os prints na consola do Eclipse dos valores de cada variável durante as diferentes iteradas para cada uma destas amostras. Analogamente os mesmos seguem em ficheiros denominados *printEM20.txt, printEM100.txt* e *printEM1000.txt*.

Como resultados estão também 3 ficheiros de Excel que possuem os pontos relativos às concentrações vs tempo, onde é possivel observar as misturas de gaussianas, e sobrepostas estão as curvas calculadas pelo algoritmo de EM. Cada curva será representada numa cor diferente e serão apresentadas de seguida:




EM100.csv
Concentração
Tempo
0
2
4
6
8
10
12
14
16
18
0
1
2
3
4
5
6
7
8
Amostra
Gaussiana 1
Gaussiana 2
Gaussiana 3
Gaussiana 4


EM1000.csv
Concentração
Tempo
0
2
4
6
8
10
12
14
16
18
0
1
2
3
4
5
6
7
8
Amostra
Gaussiana 1
Gaussiana 2
Gaussiana 3
Gaussiana 4

Como seria de esperar, não foram calculados valores de *EM3000.csv* por falta de tempo, no entanto, tentar-se-á processar os dados num computador com melhor velocidade de processamento e tentar-se-á levar resultados à discusão do projecto.

Deste modo foi possível fazer clustering de 4 diferentes comportamentos farmacocinéticos, e descobrir a expressão matemática que aproxima as concentrações relativas a cada um desses comportamentos.

Em baixo estão discriminados os parametros para cada gaussiana nas diferentes amostras.

Parametros para gaussianas EM20

a1	a2	b1	b2
75,20802	-75,208	0,543043	0,765000896
53,16345	-53,1635	0,697298	1,542997996
33,39763	-33,3976	0,332034	0,490973967
33,1993	-33,1993	0,886179	1,268959176

Parametros para gaussianas EM 1000

a1	a2	b1	b2
74,44019996	-74,4402	0,539827	0,762143895
53,32808744	-53,3281	0,698606	1,545908843
32,94241002	-32,9424	0,330703	0,488484971
33,16866499	-33,1687	0,891828	1,266668488

Parametros para gaussianas EM 100

a1	a2	b1	b2
74,80447781	-74,8045	0,542335	0,763016775
53,20726503	-53,2073	0,697447	1,542768562
32,91995977	-32,92	0,332184	0,488378467
33,19001913	-33,19	0,892785	1,266921914