Universidade de Aveiro
Dep. Electrónica Telecomunicações e Informática

Métodos Probabilísticos em Engenharia Informática

Guiões Práticos

António Teixeira Carlos Bastos Amaro Sousa

Setembro 2019

PL 1

Probabilidade: conceitos base

Palavras chave: probabilidade, experiência aleatória, espaço de amostragem, eventos, casos favoráveis, simulação, Matlab, octave.

Responda às seguintes questões utilizando sempre o Matlab ou o octave[1] para efetuar os cálculos necessários:

1. Considere a experiência aleatória de lançar **3 vezes** uma moeda equilibrada. Pretende-se calcular de forma analítica e por simulação a probabilidade de se obter **2 caras** no fim dos **3 lançamentos** e comparar os dois resultados.

   Relembre das aulas teórico-práticas que esta probabilidade é dada analiticamente pela expressão

   $$C_k^n p^k (1-p)^{n-k}$$

   em que $p = 0.5$ é a probabilidade de sair cara em cada lançamento, k é o número de caras e n é o número de lançamentos.

   Em Matlab, esta expressão é determinada da seguinte forma:

   ```
   %% Codigo 1
   p = 0.5;
   k = 2;
   n = 3;
   prob= factorial(n)/(factorial(n-k)*factorial(k))*p^k*(1-p)^(n-k)
   ```

   Por simulação, é necessário executar várias vezes a experiência aleatória de lançar 3 vezes uma moeda equilibrada e calcular a percentagem de vezes em que o resultado deu 2 caras. Em Matlab, uma forma possível de implementar este simulador é a seguinte (assumindo que a experiência é executada 10000 vezes):

   ```
   %% Código 2

   % Gerar uma matriz com 3 linhas e 10000 colunas de números aleatórios
   % entre 0.0 e 1.0 (ou seja, cada coluna representa uma experiência):
     experiencias = rand(3,10000);
   % Gerar uma matriz com 3 linhas e 10000 colunas com o valor 1 se o valor
   % da matriz anterior for superior a 0.5 (ou seja, se saiu cara) ou com o
   % valor 0 caso contrário (ou seja, saiu coroa):
     lancamentos = experiencias > 0.5; % 0.5 corresponde a 1 - prob. de caras
   % Gerar um vetor linha com 10000 elementos com a soma dos valores de cada
   % coluna da matriz anterior (ou seja, o número de caras de cada experiência):
   ```

[1] Nesta disciplina poderá utilizar o Matlab ou o octave de forma quase indistinta

```
  resultados= sum(lancamentos);
% Gerar um vetor linha com 10000 elementos com o valor 1 quando o valor do
% vetor anterior é 2 (ou seja, se a experiência deu 2 caras) ou 0 quando é
% diferente de 2:
  sucessos= resultados==2;
% Determinar o resultado final dividindo o número de experiências com 2
% caras pelo número total de experiências:
  probSimulacao= sum(sucessos)/10000
```

Note-se que o código proposto está desenvolvido passo a passo para mais fácil compreensão. Muitas das operações podem ser combinadas por forma a evitar o uso de matrizes intermédias e tornar a execução do código mais eficiente. Além disso, é útil usar variáveis iniciais para os parâmetros do problema para mais fácil alteração do código a outros casos de interesse. Assim, uma outra forma possível de implementar o mesmo simulador é a seguinte:

```
%% Código 2 - segunda versão

N= 1e5;  %número de experiências
p = 0.5; %probabilidade de cara
k = 2;   %número de caras
n = 3;   %número de lançamentos
lancamentos = rand(n,N) > p;
sucessos= sum(lancamentos)==k;
probSimulacao= sum(sucessos)/N
```

(a) Implemente os 2 métodos no Matlab.

(b) Determine a probabilidade de obter 2 caras em 3 lançamentos de uma moeda equilibrada por cada um dos métodos (execute várias vezes a simulação).

(c) Confronte os resultados dos 2 métodos.

2. Qual é a probabilidade de obter 6 caras em 15 lançamentos de uma moeda equilibrada? Confronte os resultados analíticos com os resultados de simulação.[2]

3. Qual é a probabilidade de obter pelo menos 6 caras em 15 lançamentos de uma moeda equilibrada? Confronte os resultados analíticos com os resultados de simulação.

4. Para facilitar o cálculo de outras situações similares às que tratou nos pontos anteriores, crie uma função em Matlab que permita estimar a probabilidade por simulação. A função deve ter como parâmetros de entrada p, número de lançamentos, número de caras pretendidas e número de experiências a realizar. Deve utilizar um nome adequado para a função.

   (a) Aplique a função no recálculo das questões anteriores assim como na obtenção da probabilidade para todo o espaço de amostragem quando o número de lançamentos é 20, 40 e 100.

   (b) Faça um gráfico, usando a função `stem`, das probabilidades estimadas no caso de 20 lançamentos.

5. Considere um processo de produção fabril que produz torneiras em que a probabilidade de cada torneira ser produzida com defeito é de 30%. No processo de controlo de qualidade, é selecionada uma amostra de 5 peças.

   (a) Calcule analiticamente e por simulação a probabilidade de 3 peças da amostra serem defeituosas.

   (b) Calcule analiticamente e por simulação a probabilidade de, no máximo, 2 das peças da amostra serem defeituosas.

   (c) Baseado em simulação, construa no Matlab o histograma representativo da distribuição de probabilidades do número de peças defeituosas da amostra.

[2] Adapte o Código 1 e o Código 2 de forma apropriada para resolver as questões de 2 a 4

Para revisão, responda, sozinho, às seguintes questões através de simulação e aplicando análise combinatória:

R1 Quantas sequências diferentes de 10 bits há? E de n bits?

R2 Quantas sequências diferentes de 10 símbolos do alfabeto (A,C,G,T) há? E de n símbolos do mesmo alfabeto?

R3 Um teste tem n perguntas com respostas possíveis Verdadeiro ou Falso. Forneça uma expressão para calcular o número de maneiras diferentes de responder ao teste. Qual a probabilidade de acertar todas as respostas, escolhendo-as à sorte com igual probabilidade?

R4 Quantas chaves distintas pode ter o Totoloto antigo (5 números em 49)? E o Euromilhões (5 números em 50 e duas estrelas em 11)?

R5 Considere um baralho com 20 cartas. Dessas cartas, 10 são vermelhas e numeradas de 1 a 10. As restantes 10 são pretas e também numeradas de 1 a 10.

(a) De quantas maneiras diferentes se podem dispor as 20 cartas numa fila?

(b) Calcule a probabilidade de se obter uma sequência constituída alternadamente por cartas pretas e vermelhas.

R6 Lançam-se dois dados e toma-se nota da soma de pontos obtida.

(a) Indique o espaço de amostragem (conjunto de valores possíveis) da soma.

(b) Calcule a probabilidade de se obter a soma 9.

R7 Um conjunto de 50 peças contém 8 peças defeituosas. Escolhem-se aleatoriamente 10 peças. Qual a probabilidade de encontrar 3 defeituosas?

R8 Quantas *passwords* diferentes se podem obter nas seguintes situações:

(a) comprimento 5 e cada posição contendo um dígito entre 0 e 9;

(b) comprimento 5 e cada posição contendo uma letra mínuscula sem acentos.

(c) Qual a probabilidade de acertar em cada um dos dois casos anteriores escolhendo uma password aleatoriamente ?

(d) Qual a alteração ao valor destas probabilidades de fizermos 3 tentativas completamente independentes?

PL 2

Probabilidade, probabilidade condicional, independência

Responda às seguintes questões através de simulações em Matlab/Octave e sempre que seja pedido compare com os valores teóricos:

1. Consideremos famílias com filhos:

   (a) Obtenha uma estimativa da probabilidade do acontecimento "ter pelo menos um filho rapaz" em famílias com 2 filhos. Considere que a probabilidade de nascimento de rapazes é igual à de nascimento de raparigas.

   (b) Compare com o valor obtido aplicando a teoria clássica. Que aproximações teve de efetuar? Os valores são iguais?

   (c) Suponha agora que para uma família (com os referidos 2 filhos) escolhida ao acaso sabemos que um dos filhos é rapaz. Qual a probabilidade do outro filho ser também rapaz? Obtenha o valor aplicando primeiro a definição frequencista e de seguida a teoria clássica, se aplicável.

   (d) Sabendo que o primeiro filho é rapaz, qual a probabilidade do segundo ser rapaz? O que se pode concluir do resultado obtido?

   (e) Considere uma família com mais filhos, por exemplo 5, e que sabemos que pelo menos um dos filhos é rapaz. Obtenha usando simulação uma estimativa para a probabilidade de um dos outros (e apenas um) ser também rapaz.

   (f) Repita a questão anterior, mas considerando a probabilidade de pelo menos um dos outros ser também rapaz.

2. Considere uma sequência de 10 lançamentos de uma moeda honesta em que saíram 10 caras:

   (a) Obtenha por simulação uma estimativa para a probabilidade de sair essa sequência de 10 caras?

   (b) Obtenha por simulação uma estimativa da probabilidade de sair cara no próximo lançamento, o décimo-primeiro?

   (c) Compare os valores anteriores com os valores teóricos.

3. Consideremos o problema de deteção de cancro da mama. O mamograma (como muitas outras análises clínicas) não é infalível. Estudos prolongados revelaram que:

   P["mamograma positivo se cancro da mama"] = 0,9

   P["mamograma positivo se NÃO cancro da mama"] = 0,1

   Calcule a probabilidade de uma mulher escolhida ao acaso na população portuguesa ter cancro sabendo que o seu mamograma deu positivo. Considere que a frequência de ocorrência de cancro da mama na população portuguesa feminina é de 1 em 10000.

   Estava à espera deste resultado?

   Calcule nova estimativa, mas agora considerando as mulheres que procuram a consulta específica e que para este subgrupo a ocorrência de cancro atinge 1 em 1000.

O que sugere para aumentar esta probabilidade? Se melhorar esse valor em 10% qual o ganho na probabilidade de ser mesmo cancro quando o exame dá positivo?

4. Considere o seguinte "jogo": lançamento com os olhos vendados de $n = 20$ dardos, um de cada vez, para $m = 100$ alvos, garantindo-se que cada dardo atinge sempre um alvo (e apenas 1).

   a) Qual a probabilidade de nenhum alvo ter sido atingido mais do que uma vez?

   b) Qual a probabilidade de pelo menos 1 alvo ter sido atingido 2 (ou mais) vezes?

   c) Faça gráficos da variação em função de n do valor da probabilidade da alínea b). Considere para $m = 1000, 10000, 100000, 1000000$ e para cada valor de m valores de n entre e 1 e 100, com incremento de 10. Os 4 gráficos devem ser sub-gráficos de uma mesma figura.

   d) Faça um gráfico similar em função de m, fixando n em 100.

5. ** TPC ** Considere um array de tamanho 1000 que serve de base à implementação de uma memória associativa (por exemplo em Java) e que se pode assumir que a função de hash devolve um valor entre 0 e 999 com igual probabilidade.

   (a) Qual a probabilidade de haver colisões (pelo menos 2 *keys* mapeadas pela função de hash para a mesma posição do array) em situações em que temos apenas 10 *keys* ?

   (b) Faça um gráfico da probabilidade de colisões em função do número de keys.

   (c) Para um número fixo de keys, por exemplo 50, represente graficamente a variação da probabilidade de não haver qualquer colisão em função do tamanho do array.

6. Consideremos uma festa em que estão presentes um determinado número de pessoas, que designamos por n.

   (a) Qual deve ser o menor valor de n para o qual a probabilidade de duas ou mais pessoas terem a mesma data de aniversário (mês e dia) é superior a 0,5?

   (b) Qual deve ser o valor de n para que a probabilidade da alínea anterior passe a ser superior a 0,9?

7. You have a fair five-sided die. The sides of the die are numbered from 1 to 5. Each die roll is independent of all others, and all faces are equally likely to come out on top when the die is rolled. Suppose you roll the die twice.

   (a) Let event A to be "the total of two rolls is 10", event B be "at least one roll resulted in 5", and event C be "at least one roll resulted in 1".

   i. Is event A independent of event B?

   ii. Is event A independent of event C?

   (b) Let event D be "the total of two rolls is 7", event E be "the difference between the two roll outcomes is exactly 1", and event F be "the second roll resulted in a higher number than the first roll".

   i. Are events E and F independent?

   ii. Are events E and F independent given event D?

8. Considere uma linguagem com apenas 3 palavras {"um", "dois", "três"} e que permite sequências de 2 palavras. Se todas as frases forem equiprováveis e as duas palavras na frase puderem ser iguais:

   (a) Qual a probabilidade da sequência "um dois"?

   (b) Qual a probabilidade de "um" aparecer pelo menos uma vez?

   (c) Qual a probabilidade de ocorrer "um" ou "dois"?

   (d) Qual o valor de P["sequência incluir a palavra um" | "sequência inclui palavra dois"]?

   (e) Resolva a questão anterior para o caso de termos 10 palavras diferentes e sequências de 5 palavras com ajuda de um programa que calcule exaustivamente todas as possibilidades. Sugestão: use números de 1 a 10 para representar as palavras e use Matlab/Octave. Tente criar uma função com os parâmetros de entrada que considere adequados.

(f) Repita a questão anterior para 11, 12, ... 20 palavras diferentes e represente a variação da probabilidade num gráfico. Nota: Devido à memória necessária não se pode ter valores muito elevados.

(g) Adicione ao gráfico anterior, para comparação, a probabilidade se a linguagem não permitir repetições das palavras.

9. Considere que uma empresa tem 3 programadores (André, Bruno e Carlos) e que a probabilidade de um programa de cada um deles ter problemas ("bugs") e o número de programas desenvolvidos assumem os valores apresentados na tabela seguinte.

Programador	Prob("erro num programa")	programas
André	0.01	20
Bruno	0.05	30
Carlos	0.001	50

O Diretor da empresa seleciona de forma aleatória um dos 100 programas produzidos pelos seus 3 programadores e descobre que este contém um erro sério.

(a) Qual é a probabilidade de o programa ser do Carlos?

(b) De quem é mais provável ser o programa?

Responda à questão seguinte usando uma abordagem teórica:

10. Most mornings, Victor checks the weather report before deciding whether to carry an umbrella. If the forecast is "rain" the probability of actually having rain that day is 80%. On the other hand, if the forecast is "no rain", the probability of it actually raining is equal to 10%. During fall and winter the forecast is "rain" 70% of the time and during summer and spring it is 20%.

(a) One day, Victor missed the forecast and it rained. What is the probability that the forecast was "rain" if it was during the winter? What is the probability that the forecast was "rain" if it was during the summer?

(b) The probability of Victor missing the morning forecast is equal to 20 % on any day in the year. If he misses the forecast, Victor will flip a fair coin to decide whether to carry an umbrella. On any day of a given season he sees the forecast, if it says "rain" he will always carry an umbrella, and if it says "no rain", he will not carry an umbrella. Are the events "Victor is carrying an umbrella", and "The forecast is no rain" independent? Does your answer depend on the season?

(c) Victor is carrying an umbrella and it is not raining. What is the probability that he saw the forecast? Does it depend on the season?

PL 3

Variáveis Aleatórias

Palavras chave: Variáveis Aleatórias: distribuição de probabilidade, função massa de probabilidade, função de probabilidade, função distribuição acumulada, função densidade de probabilidade, e algumas distribuições (Binomial, Poisson, Uniforme, Normal).

Responda às seguintes questões:

1. Considere a variável aleatória X correspondente à face que fica visível no lançamento de 1 dado:

   (a) Efectue em Matlab/Octave um gráfico representativo da função de massa de probabilidade[1] de X. Não se esqueça de ter os valores adequados no eixo dos xx;

   (b) Num segundo gráfico, na mesma figura, desenhe o gráfico da função de distribuição acumulada.

2. Considere uma caixa contendo 90 notas de 5 Euros, 9 notas de 50 e 1 de 100:

   (a) Descreva o espaço de amostragem da experiência aleatória, retirar uma nota da caixa, e as probabilidades dos acontecimentos elementares.

   (b) Considere a variável aleatória X como sendo o valor de uma nota retirada à sorte da caixa acima descrita. Descreva o espaço de amostragem e a função de massa de probabilidade de X.

   (c) Determine a função massa de probabilidade de X e efectue a sua representação gráfica em Matlab/Octave.

3. Considere 4 lançamentos de uma moeda equilibrada. Seja X a variável aleatória representativa do número de coroas observados nos 4 lançamentos.

   (a) Estime, usando simulação com o Matlab/octave, a função de probabilidade $p_X(x)$ da variável aleatória X.

   (b) Estime o valor esperado, a variância e o desvio padrão de X com base em $p_X(x)$.

   (c) Identifique o tipo da distribuição da variável aleatória X e escreva a expressão teórica da respectiva função de probabilidade.

   (d) Substitua os valores admissíveis da v.a. X na função obtida acima e compare com os cálculos efectuados na alínea a) desta questão.

   (e) Com base na função de probabilidade desta distribuição, calcule:

      i. A probabilidade de obter pelo menos 2 coroas;

      ii. A probabilidade de obter até 1 coroa;

      iii. A probabilidade de obter entre 1 e 3 coroas.

4. Sabendo que um processo de fabrico produz 30% de peças defeituosas e considerando a variável aleatória, X, que representa o número de peças defeituosas numa amostra de 5 peças tomadas aleatoriamente, obenha (analiticamente e por simulação):

[1] A função massa de probabilidade pode ser designada alternativamente por função de probabilidade

(a) o histograma representativo da distribuição de probabilidades de X;

(b) a probabilidade de, no máximo, 2 das peças de uma amostra serem defeituosas.

5. Suponha que o(s) motor(es) de um avião pode(m) falhar com probabilidade p e que as falhas são independentes entre motores. Suponha ainda que o avião se despenha se mais de metade dos motores falharem. Utilize a distribuição que considerar mais adequada.

   a) Prefere voar num avião com 2 ou 4 motores?

   Sugestão: Tem pelo menos 2 alternativas: (1) obter expressões para a probabilidade de cada tipo de avião se despenhar em função de p e usar o quociente entre ambas para responder à questão, (2) efectuar os cálculos para um conjunto de valores concretos[2] de p (ex: `p= logspace(-3,log10(1/2),100)`) e usar um gráfico mostrando simultaneamente as probabilidades de cada tipo de avião se despenhar.

6. A distribuição de Poisson é uma forma limite da distribuição binomial (quando $n \to \infty$, $p \to 0$ e np permanece constante) e portanto pode ser usada para aproximar e simplificar os cálculos envolvidos com a binomial numa situação dessas.

   Num processo industrial de fabrico de chips, alguns aparecem defeituosos tornando-os inapropriados para comercialização. É sabido que em média por cada mil chips há um defeituoso. Determine a probabilidade de numa amostra de 8000 aparecerem 7 defeituosos. Compare os resultados usando a distribuição correcta e a aproximação de Poisson.

   Lei de Poisson: $p_k = \frac{\alpha^k}{k!} e^{-\alpha}$

7. É conhecido que o número de mensagens que chega a um computador por segundo se comporta de acordo com a lei de Poisson. Suponha que o número de mensagens que chega a um computador segue uma lei de Poisson com média 15 (por segundo). Calcule a probabilidade de:

   (a) o computador não receber nenhuma mensagem num segundo.

   (b) mais de 10 mensagens chegarem ao computador num período de um segundo.

8. Verifique se a função $f(x) = (x + 5)/30$ pode representar a função de probabilidade de uma variável aleatória que só possa tomar os valores 1, 2, 3 e 4.

9. Assumindo que o número de erros tipográficos numa página de um livro tem uma distribuição de Poisson com $\lambda = 1$, calcule a probabilidade de que exista pelo menos um erro numa determinada página.

10. Sendo a variável aleatória X contínua e uniformemente distribuída em $(0, 10)$, calcule as probabilidades de:

    (a) $X < 3$

    (b) $X > 7$

    (c) $1 < X < 6$

    Comprove os resultados através de simulação.

11. Considerando a variável aleatória X, representativa das classificações dos alunos de um determinado curso, contínua [3] e com distribuição normal (média 14 e desvio padrão 2), obtenha através de simulação uma estimativa para as probabilidades de:

    (a) um aluno do curso ter uma classificação entre 12 e 16;

    (b) os alunos terem classificações entre 10 e 18;

    (c) um aluno passar (ter classificação maior ou igual a 10).

    Sugestão: Utilize a função Matlab `randn()` para gerar números aleatórios com distribuição normal.

[2] Correr `help logspace` no Matlab/Octave para perceber os argumentos do `logspace` usados no exemplo.

[3] Equivale a considerar que as classificações são números reais.

PL 4

Geração de números aleatórios

Palavras-chave: geração de números aleatórios, geração de variáveis aleatórias.

Consulte a informação das aulas Teórico-Práticas e responda às questões seguintes através de pequenos programas em Octave/Matlab:

1. Implemente uma função que gere um vector de números aleatórios com base no método da Congruência.

   A função deve permitir controlar o seu comportamento através dos parâmetros necessários à definição da fórmula usada neste método e também permitir definir o comprimento do vector a gerar. Sugestão: `function y = lcg (X0,a,c,m,N).`

   Utilizando a função:

   (a) gere um vector de comprimento 1000 usando parâmetros à sua escolha e visualize o seu histograma. Quantos números diferentes obteve? Sugestão: experimente a função `unique()`.

   (b) gere um conjunto de números aleatórios entre 0 e 1 com base nos números gerados na alínea a). Visualize o histograma e conte novamente os números diferentes.

   (c) repita as alíneas anteriores com os parâmetros utilizados na implementação incluída na biblioteca NAG (ver apresentação da TP).

2. Usando a função Matlab/Octave `rand()` , que permite de uma forma simples obter números aleatórios com características semelhantes aos que obtivemos anteriormente:

   (a) Simule uma sequência de 10 experiências de Bernoulli com probabilidade de sucesso p. Represente sucesso por 1 ;

   (b) Simule 15 lançamentos de 1 dado (honesto);

   (c) Obtenha um conjunto de 20 números reais igualmente distribuídos entre -4 e 10;

3. Crie uma função que gere um vector de números com uma distribuição de Bernoulli com parâmetro p.

   Utilize o histograma (função `hist()`) e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento.

   Exemplo de teste: hist(Bernoulli(.3, 10000),(0:1)')

4. Crie uma função que gere um vector de números com uma distribuição Binomial, dados os parâmetros deste tipo de distribuição (n e p) e N (tamanho do vector a gerar).

   Utilize o histograma e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento. Tente comparar as probabilidades obtidas com base no vector gerado com os valores teóricos.

5. Crie uma função que gere um vector de números com uma distribuição discreta genérica definida pela sua função de massa de probabilidade (fmp).

   A sua função deve receber como parâmetros de entrada dois vectores definindo a fmp, xi e pX, assim como o número de valores a gerar.

Utilize o histograma e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento. Sugere-se que para o primeiro teste se use uma fmp de um dado nada honesto em que a probabilidade de sair 6 é bem maior do que a probabilidade de sairem as outras faces, não existindo diferenças de probabilidade entre as outras faces.

6. Utilize o método da transformação para implementar uma função Matlab/Octave capaz de gerar números com uma distribuição exponencial. Use exponencial para o nome dessa função.

   Teste a função, usando, por exemplo `hist(exponencial (10000,.5),20)`.

7. Crie uma função que gere um vector de números com a distribuição normal normalizada utilizando o Método de Box-Müller.

   Usando esta função como base, gere as classificações de uma turma de 30 alunos por forma a terem média 14 e variância 2.

   Confirme os resultados com a função Matlab `randn()` . Consulte a informação sobre esta função.

8. Repita o exercício anterior criando uma nova versão da função baseada no método da rejeição. Assuma que os valores de $f_X(x)$ da função Gaussiana normalizada é próxima de zero fora do intervalo (-5,5).

   Use o histograma para verificar as características principais (forma, média e dispersão em torno da média).

9. (TPC) Crie uma função que permita simular a retirada aleatória de um subconjunto de bolas de uma urna, sem reposição.

   Aplique a função ao caso do Totoloto.

PL 5

Funções de dispersão / hash functions

Palavras-chave: aplicações de métodos probabilísticos, simulação, *hash table* , *hash function*, *chaining hash table*

O objectivo deste exercício é avaliar o funcionamento de uma estrutura de dados importante, a *Chaining Hash Table*, e um dos conceitos que a suportam, *Hash Functions* (funções de dispersão).

Sabe-se que "o desempenho da tabela de dispersão depende da capacidade da função de hash para distribuir uniformemente as chaves pelos índices do array" e que "uma análise estatística da distribuição das chaves pode ser considerada". Neste exercício, usando geração de chaves de forma aleatória iremos efectuar a análise da distribuição das chaves.

Começaremos por criar os componentes essenciais para efectuar uma primeira simulação bastante simplificada. De seguida tornaremos um pouco mais realista.

- Primeira versão:

  1. Crie uma função que gere uma chave (string) com comprimento aleatório entre 3 e 20 assumindo uma distribuição uniforme (discreta) e em que as letras (apenas minúsculas e maiúsculas) são equiprováveis.
  2. Implemente uma função de hash simples.Sugestão: Use uma das seguintes funções de dispersão:
     `string2hash()`[1],
     `hashstring()`[2],
     `DJB31MA()`[3],
     `hashcode()`[4],
     Nota: as 3 últimas funções são apresentadas noutras linguagens, sendo um bom exercício adaptá-las para o Matlab/octave e tentar perceber as relações entre elas. Tenha em atenção a necessidade de implementar estas funções em aritmética inteira.
  3. Simule a inserção de 1000 das strings criadas pela função que desenvolveu numa *Chaining Hash Table* (que como se deve recordar usa internamente um array e listas ligadas). Como para a nossa simulação não necessitamos guardar as chaves e valores a elas associados, apenas necessitamos do array. Adopte para tamanho do array um valor que implique um factor de carga elevado (ex: 0.8) .
     Guarde informação sobre o número de chaves que foram mapeadas numa determinada posição do array até esse momento.
     Durante a simulação, visualize, em gráficos separados:
     - o número de strings que foram mapeadas pela hash function para cada posição;
     - o histograma desses números.

[1] `https://www.mathworks.com/matlabcentral/fileexchange/27940-string2hash`
[2] Funçãao usada em Programação II que poderá ser adaptada ao matlab/octave.
[3] Função baseada no algorítmo de Daniel J. Bernstein, ver sumário de MPEI de 2014 (Prof. Paulo Jorge Ferreira) ou slides TP.
[4] Implementação do método hashcode, em JAVA, para objetos do da classe String

4. Usando a informação sobre o número de chaves que foram mapeadas para cada posição do array após a inserção de todas as chaves, estime e represente graficamente a função de distribuição para a variável aleatória X definida como o número de chaves mapeadas para uma posição. Qual seria o valor médio do comprimento das listas ligadas neste caso ? Qual a variância ?

- Segunda versão:

  1. Crie uma nova versão da função de geração das chaves assumindo que o tamanho das chaves segue uma distribuição normal (com média 10 e variância 5) e as letras seguem a distribuição das letras em Português.

     Nota: Utilize o script disponibilizado no material de apoio à UC para efectuar a estimativa dessa distribuição.
  2. Repita a simulação com 1000 chaves e os cálculos da última alínea da versão 1.

- Terceira versão (opcional):

  1. Altere a função de hash na segunda versão e repita a simulação e cálculos.

- Questões finais:

  A função de hash conseguiu o objectivo de espalhar as chaves/strings pelo array ? Temos de facto uma distribuição uniforme das chaves pelos índices do array ?

PL 6

Filtros de Bloom

Palavras-chave: Estruturas de dados probabilísticas, Filtros de Bloom (*Bloom Filters*), Funções de dispersão.

Este trabalho prático tem por objectivo criar e testar um módulo que suporte a criação de *Bloom filters* (ex: [2, Cap. 4: Mining Data Streams]). Para tal, execute os seguintes passos:

1) Crie, em Matlab, um conjunto de funções que implementem as funcionalidades de um *Bloom Filter* básico. As funções devem ter os parâmetros necessários para que seja possível criar *Bloom filters* de diferentes dimensões e usando números diferentes de funções de dispersão (k).

   Sugestão 1: Criar pelo menos 3 funções [1, sec. 3.2]: uma para inicializar a estrutura de dados; outra para inserir um elemento (ou elementos) no filtro; uma terceira para verificar se um elemento pertence ao conjunto.

   Sugestão 2: Deve procurar, seleccionar e implementar uma função de dispersão que tenha bom desempenho [1]

2) Teste as funções que desenvolveu na criação de um pequeno *Bloom Filter* para guardar uma lista de países. Insira alguns nomes de países no filtro e teste a pertença desses e de alguns países adicionais que não pertençam a essa lista inicial.

3) Para um teste mais exaustivo:

   (a) Gere m=1000 strings aleatórias com 40 caracteres (considere como caracteres possíveis o conjunto de caracteres minúsculos, maiúsculos e algarismos) e preencha um *Bloom Filter*, de tamanho n=8000. Este *Bloom Filter* deve ter $k = 3$ funções de dispersão e as strings geradas devem obdecer às diferentes probabilidades de ocorrência das letras em português.

   (b) Gere um segundo conjunto de 10000 strings aleatórias com 40 caracteres e teste a pertença das mesmas ao *Bloom Filter* que preencheu antes. Quantas destas strings foram consideradas como pertencendo ao conjunto com que o filtro foi preenchido? Estava à espera deste resultado?

4) Repita o teste da questão anterior para um número diferente de funções de dispersão ($k = 1, \ldots, 15$), obtendo o número de falsos positivos para cada k. Represente num gráfico os valores obtidos, em função de k e sobreponha nesse gráfico os valores teóricos (Assuma a independência de hash functions e que cada uma seleciona cada posição do bloom filter com igual probabilidade). Nota: Assume-se que as 10000 strings de teste são todas diferentes das 1000 inseridas no *Bloom filter*. No entanto pode haver strings iguais.

5) [opcional] Seleccione dois textos com um número grande de palavras de `gutemberg.org` e utilize um Bloom Filter para determinar as palavras do segundo que não existem no primeiro.

[1] Nota Importante: É obrigatório manter a informação original sobre autores e afins em todas as funções que utilizar e que não sejam da vossa autoria. Adaptações de código existente apenas podem ser feitas se as condições de utilização definidas pelos autores o permitirem, mantendo sempre informação sobre o autor original e adicionando informação sobre quem fez a alteração/evolução. Neste trabalho sugere-se criação de código original para todas as funções com a excepção das funções de dispersão.

Existe possibilidade de alguma das palavras que identificou como não existentes no primeiro texto fazerem parte desse texto?

6) Adapte as suas funções para implementar um *Count Filter*. Aplique estas novas funções para conseguir mostrar para uma qualquer palavra de um livro o número de vezes que ocorre no livro. Esta contagem apenas deve ser mostrada para palavras que pertençam ao livro.

   Sugere-se a utilização de um livro do projecto Gutemberg.

PL 7

Similaridade de conjuntos: minHash

Palavras-chave: Similaridade de conjuntos, distância de Jaccard, indíce de Jaccard, minhash.

O presente trabalho prático tem por objectivo criar um "módulo" que suporte a descoberta de conjuntos similares (ex: [2, Cap. 3: Finding Similar Items]) e testar esse módulo.
Neste trabalho iremos utilizar o ficheiro `u.data` do conjunto de dados (release 4/1998) MovieLens 100k, disponível em `http://grouplens.org/datasets/movielens/`. Este ficheiro contém informação sobre 943 utilizadores e 1682 filmes. Tem cerca de 100 000 linhas, como as seguintes:

```
196    242    3    881250949
186    302    3    891717742
 22    377    1    878887116
```

As colunas são separadas por *tabs*; a primeira coluna contém o ID do utilizador; a segunda contém o ID de um filme (avaliado pelo utilizador mencionado na primeira coluna); a terceira é a avaliação; a quarta um *timestamp*.

O nosso objectivos é descobrir utilizadores que avaliaram conjuntos similares de filmes. Para este objectivo as colunas 3 e 4 não são necessárias.

1) Analise o código Matlab disponibilizado conjuntamente com este guião e complete-o por forma a conseguir calcular a **distância de Jaccard** entre os conjuntos de filmes avaliados pelos vários utilizadores.

   Inclua no código a possibilidade de calcular o tempo que demora cada uma das partes (cálculo da distância e determinação das distâncias abaixo de um determinado limiar). Veja a informção relativa a `tic, toc, cputime, etime`.

   No final, o programa deve mostrar informação com: (1) número de pares de utilizadores com distâncias inferiores ao limiar definido; (2) informação sobre cada par (utilizadores e distância).

   Adicione, também, a capacidade de gravar em ficheiro a matriz de distâncias calculada. Sugere-se que consulte a informação de `save`.

2) Com base no código que adaptou, crie funções para:

   - Criar a estrutura de dados com os conjuntos de filmes;
   - Calcular as distâncias ;
   - Processar as distâncias e devolver os items similares. Esta função deve ter como um dos parâmetros o limiar de decisão.

   Teste o código com 100 utilizadores seleccionados de forma aleatória.

   Depois do teste anterior com um número reduzido de utilizadores e eventual resolução de problemas detectados, execute o seu programa com todo o conjunto de dados por forma a determinar todos os pares de utilizadores com uma distância de Jaccard inferior a 0.4

   Tome nota dos tempos e dos resultados obtidos.

3) Crie uma nova versão da função de cálculo de distância recorrendo a uma aproximação probabilística usando minhash. Comece por testar esta nova implementação com um número pequeno de utilizadores e depois teste-a com o conjunto total de utilizadores.

   Compare os pares considerados como similares com os obtidos com a implementação não probabilística. Comente.

4) Adapte o seu código para utilizar outro(s) conjunto(s) de dados (datasets) do Movilens e utilize as funções desenvolvidas anteriormente para detetar os pares de utilizadores similares.

```
% Código base para guião PL07 MPEI 2018-2019

udata=load('u.data');  % Carrega o ficheiro dos dados dos filmes
% Fica apenas com as duas primeiras colunas
u= udata(1:end,1:2); clear udata;

% Lista de utilizadores
users = unique(u(:,1));     % Extrai os IDs dos utilizadores
Nu= length(users);          % Número de utilizadores

% Constrói a lista de filmes para cada utilizador
Set= cell(Nu,1);            % Usa células
for n = 1:Nu,  % Para cada utilizador
    % Obtém os filmes de cada um
    ind = find(u(:,1) == users(n));
    % E guarda num array. Usa células porque utilizador tem um número
    % diferente de filmes. Se fossem iguais podia ser um array
    Set{n} = [Set{n} u(ind,2)];
end
%% Calcula a distância de Jaccard entre todos os pares pela definição.

J=zeros(...);  % array para guardar distâncias
h= waitbar(0,'Calculating');
for n1= 1:Nu,
    waitbar(n1/Nu,h);
    for n2= n1+1:Nu,
           %%  Adicionar código aqui
    end
end
delete (h)
%%  Com base na distância, determina pares com
%%  distância inferior a um limiar pré-definido

threshold =0.4  % limiar de decisão

% Array para guardar pares similares (user1, user2, distância)
SimilarUsers= zeros(1,3);
k= 1;
for n1= 1:Nu,
    for n2= n1+1:Nu,
        if  % ......
            SimilarUsers(k,:)= [users(n1) users(n2) J(n1,n2)]
            k= k+1;
        end
    end
end
```

PL 8

Cadeias de Markov

Palavras-chave: Cadeias de Markov, matriz de transição, múltiplas transições, estado estacionário, estados absorventes, tempo até à absorção, simulação.

Nota: Adopte a a definição da matriz de transição (de estados) em que o elemento t_{ij} da matriz corresponde à probabilidade de transição do estado j para o estado i.

1. Considere a seguinte situação e responda às alíneas abaixo:

   Um aluno do primeiro ano de um curso de Engenharia tem todas as semanas 2 aulas Teórico-Práticas de uma Unidade Curricular X às 9:00, às quartas e sextas.

   Todos os dias que tem aulas desta UC, decide se vai à aula ou não da seguinte forma: Se tiver estado presente na aula anterior a probabilidade de ir à aula é 70 %; se faltou à anterior, a probabilidade de ir é 80 %.

   (a) Se estiver presente na aula de quarta numa determinada semana, qual a probabilidade de estar presente na aula de quarta da semana seguinte ?

   Sugestão: Comece por definir a matriz de transição e o vetor estado correspondentes.

   (b) Se não estiver presente na aula de quarta numa determinada semana, qual a probabilidade de estar presente na aula de quarta da semana seguinte ?

   (c) Sabendo que esteve presente na primeira aula, qual a probabilidade de estar na última aula, assumindo que o semestre tem exactamente 15 semanas de aulas e não existem feriados?

   (d) Represente num gráfico a probabilidade de faltar a cada uma das 30 aulas, assumindo que a probabilidade de estar presente na primeira aula é de 85 %.

2. Considere a seguinte "dança" de grupos: Divide-se uma turma em 3 grupos (A, B e C) no início do semestre e no final de cada aula efectuam-se os seguintes movimentos:

   - $1/3$ do grupo A vai para o grupo B e outro $1/3$ do grupo A vai para o grupo C;
   - $1/4$ do grupo B vai para A e $1/5$ de B vai para C
   - Metade do grupo C vai para o grupo B; a outra mantém-se no grupo C.

   (a) Crie em Matlab a matriz de transição.

   Confirme que se trata de uma matriz estocástica.

   (b) Crie o vector relativo ao estado inicial considerando que no total temos 90 alunos, o grupo A tem o dobro da soma dos outros dois e os grupos B e C têm o mesmo número de alunos.

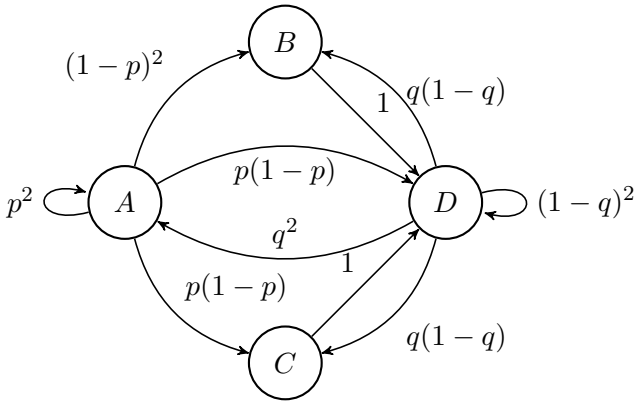
   (c) Quantos elementos integrarão cada grupo no fim da aula 30 considerando como estado inicial o definido na alínea anterior?

   (d) Quantos elementos integrarão cada grupo no fim da aula 30 considerando que inicialmente se distribuiram os 90 alunos equitativamente pelos 3 grupos?

3. Crie uma matriz de transição para uma cadeia de 20 estados gerando os elementos dessa matriz com a ajuda da função rand(). Com base nessa matriz:

   (a) Qual a probabilidade de o sistema fazer uma transição entre o primeiro e o último estado em 20 transições ? E em 40 ? E em 100 ?

4. Considere o seguinte diagrama representativo de uma Cadeia de Markov:



   (a) Defina, em Matlab, a matriz de transição T. Assuma $p = 0,4$ e $q = 0,6$;

   (b) Qual a probabilidade de o sistema chegar ao estado B após 10 transições adicionais caso inicialmente se encontre no estado A ?
   E de chegar a cada um dos outros estados para as mesmas condições ?

   (c) Visualize o comportamento desta cadeia usando o Markov chain "playground", disponível em `http://setosa.io/markov/index.html`.

5. Suponha que observa o estado do tempo uma vez por dia (por exemplo, de manhã às 11:00) e que considera três estados possíveis: sol, nuvens e chuva. Assumindo que o tempo no dia $n + 1$ apenas depende do tempo no dia n e que as probabilidades de transição são as da tabela seguinte, responda às questões abaixo:

dia n \ dia $n+1 \rightarrow$	sol	nuvens	chuva
sol	0,7	0,2	0,1
nuvens	0,2	0,3	0,5
chuva	0,3	0,3	0,4

   (a) Defina, em Matlab, a correspondente matriz de transição;

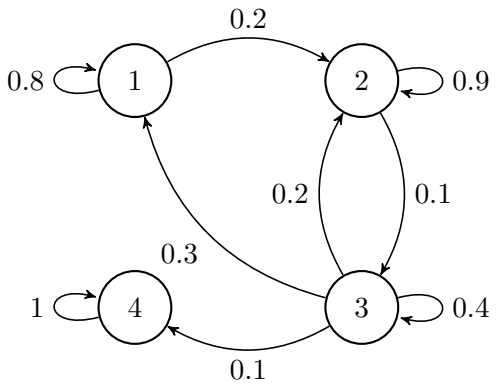
   (b) Assumindo que a observação inicial (digamos no dia 0) é que o dia é de sol, qual a probabilidade do dia 2 ser de chuva ?

   (c) Calcule as n primeiras potências de T (n=20) e apresente num gráfico a evolução dos vários elementos da matriz em função de n;

   (d) Repita o processo da alínea anterior parando-o assim que o máximo do módulo da diferença entre os valores dos elementos da matriz em duas iterações consecutivas não exceda 10^{-4};

   (e) ** TPC ** Visualize o comportamento desta cadeia usando o Markov chain "playground", disponível em `http://setosa.io/markov/index.html`.

6. Considere o seguinte conjunto de páginas web ligadas entre si:



   (a) Escreva a matriz de transição H (de Hyperlinks), com H_{ji} sendo a probabilidade de ir da página i para a página j num único passo. Crie em Matlab/Octave essa matriz.

   (b) Qual a probabilidade de começando na página 1 ao fim de 1000 passos estar na página 2? Estava à espera deste valor?

   (c) Determine a probabilidade de chegar à página j a partir da página i, em 1,2,10 e 100 passos.

   (d) Determine a matriz Q.

   (e) Determine a matriz fundamental F.

   (f) Qual a média (valor esperado) do número de passos necessários para atingir a página 4 começando na página 1? e se começarmos na página 2? e se iniciarmos na página 3?

   (g) Qual o tempo até à absorção das páginas 1 a 3?

   (h) Modifique a matriz H para aumentar esse tempo (até à absorção) e recalcule Q, F e o tempo até à absorção.

   (i) Confirme os valores dos pontos anteriores através de simulação (faça a média de várias simulações). Use o código Octave no verso como base para criar a suas simulações.

7. Três amigos, a Ana, o Bernardo e a Catarina, resolveram trocar dinheiro entre si uma vez por dia durante um ano da seguinte maneira: A Ana dá 20% do que tem ao Bernardo e fica com o resto para si. O Bernardo dá 10% do que tem à Ana, dá 30% do que tem à Catarina e fica com o resto para si. A Catarina dá 5% do que tem à Ana, dá 20% do que tem ao Bernardo e fica com o resto para si. O dinheiro é transferido electronicamente, sem arredondamento, às 23h59m de cada dia e é creditado na conta de cada um no início do dia seguinte.

   Sabendo que às 12h do dia 1 de Janeiro de 2015 a Ana tinha 100 euros, que o Bernardo tinha 200 euros e que a Catarina tinha 30 euros, responda às seguintes questões:

   (a) Às 12h do dia 4 de Janeiro, quanto dinheiro tinha cada um dos amigos?

   Resposta: Ana _____
   Bernardo _____
   Catarina _____

   (b) Logo depois da passagem de ano para o ano de 2016, com quanto dinheiro vai ficar cada um dos amigos?

   Resposta: Ana _____
   Bernardo _____
   Catarina _____

   (c) Em que dia, no formato dia do mês e mês, passa a Catarina a ter mais de 130 euros?

   Resposta: _____

```
# an example state transition matrix (page 3 is absorving)
H = [0.9 0.5 0 ;
     0.1 0.4 0 ;
     0   0.1 1 ];
# the fundamental matrix
Q = H(1:2,1:2);
F = inv(eye(2)-Q)

# given a transition matrix and the current state,
# this function returns the next state
function state = nextState(H, currentState)
     # find the probabilities of reaching all pages starting at the current one
     probVector = H(:,currentState);  # Attention: it is a column vector
     # n is the number of pages, that is, H is n x n
     n = length(probVector);
     # pick the next page randomly according to those probabilities
     state = discrete_rnd(1:n, probVector);
endfunction

# random walk on the graph according to state transition matrix H
# first = initial state, last = terminal or absorving state
function state = crawl(H, first, last)
     # the sequence of states will be saved in the vector "state"
     # initially, the vector contains only the initial state
     state = [first];
     # keep moving from page to page until page "last" is reached
     while (1)
          state(end+1) = nextState(H, state(end));
          if (state(end) == last) break; endif
     endwhile
endfunction

# pick the next page randomly according to those probabilities
# states = vector with states (numbers), probVector = probability vector
function state = discrete_rnd(states, probVector)
#...  To be developed


# how to use crawl()
state = crawl(H, 1, 3);
```

Bibliografia

[1] James Blustein and Amal El-Maazawi. Bloom filters - a tutorial, analysis, and survey. Tecnhical Report CS-2002-10, Dalhousie University, Dec 2002.

[2] Jure Leskovec, Anand Rajaraman, and Jeff Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.