Métodos Probabilísticos para Engenharia Informática

2018-2019

Aula 1

Informações sobre a cadeira

(Revisões de) probabilidades

Informações sobre a cadeira

Objectivos da cadeira

- Desenvolver a capacidade de aplicar métodos probabilísticos em engenharia informática
  - Suportada no conhecimento de conceitos essenciais
- Complementar a formação em métodos determinísticos
  - (da generalidade das outras UCs do MIECT e LEI)

Funcionamento da cadeira

- TPs (2x1.5 h) + PL (2 h) por semana

Teórico-Práticas:

- **Noções básicas de probabilidade**
- **Simulação**
- **Variáveis aleatórias e distribuições**
- **Aplicações representativas**
- **Cadeias de Markov**

Aulas Práticas

- 7/8 guiões para 1 (ou 2 aulas)
  - PL1 – Probabilidades, Probabilidade condicional
  - PL2 – Variáveis aleatórias
  - PL3 – Geração de Números aleatórios e Simulação
  - Aplicações (Bloom filters, …
  - …
  - Cadeias de Markov

OT

- Por marcação, email para o respectivo docente até às 12 horas do dia da OT:
  - Terça-feira, Prof. António Teixeira, ajst@ua.pt
  - Quinta-feira, Prof. Carlos Bastos, cbastos@ua.pt
- Alternativamente, marcação direta com os respectivos docentes.

Faltas

- Há marcação de faltas nas TPs e nas PLs nos termos do Regulamento de Estudos da Universidade de Aveiro https://www.ua.pt/sga/PageText.aspx?id=4646

Equipa docente 2018-2019

- Carlos Bastos (cbastos@ua.pt)
  Regente
  - TP2, 2ª e 5ª
  - PL P2 (2ª)
  - OT2 (5ª)

- António Teixeira (ajst@ua.pt)
  - TP1, 2ª e 3ª
  - PLs P3 (3ª) e P4 (5ª)
  - OT1 (3ª)

- Armando Pinho (ap@ua.pt)
  - PL P6 (2ª)

- Daniel Castanheira (dcastanheira@av.it.pt)
  - PLs´, P5 e P7 (5ª)

Avaliação

- **Avaliação discreta**:
  - 20 % **TP** (exame escrito a realizar em data a anunciar, em princípio, na 3ª ou 4ª semanas de outubro)
  - 20 % **P** (avaliação do desempenho nos trabalhos das aulas práticas)
  - 30 % **P** (mini projecto e sua apresentação)
  - 30 % **P** (mini teste prático em computador)
    - Época de exames

Métodos probabilísticos para cursos de Engª. de Computadores e de Engª Informática?

Probabilidades para Informática ?

- **Muitos problemas** na área da Informática, Ciências da Computação e afins **contêm algum grau de aleatoriedade**

- Exemplos:
  - Quantos computadores estarão ligados ao longo do dia a uma determinada rede wireless?
  - Qual a palavra mais provável que um utilizador irá escrever ao escrever um SMS?
  - Quais as páginas da web que têm mais relevância para uma procura ?

Probabilidades para Informática ?

- Também se podem **resolver** muitos problemas usando abordagens não determinísticas …
  - Muitas vezes com vantagens em termos de, por exemplo, velocidade

Exemplos de Aplicação

- Algoritmos probabilísticos
  - Ordenação, Métodos de Monte Carlo e Las Vegas
- Simulação
  - Redes de dados, ataques informáticos …
- Análise probabilística de algoritmos
- Teste de Software
- Poupança de memória
  - Ex: Bloom filters, contadores aleatórios
- Programação probabilística

Algoritmos probabilísticos

- Algoritmos que efetuam decisões aleatórias durante a sua execução

- Exemplo: Quicksort com pivot decidido de forma aleatória

- Vantagens:
  - Para muitos problemas um algoritmo probabilístico é o mais simples, o mais rápido, ou ambos

Algoritmos probabilísticos:

exemplos áreas de aplicação

- Teoria de números
  - Teste de números primos
- Estruturas de dados
  - Procura, ordenação, geometria computacional…
- Identidades algébricas
  - Verificação de matrizes e polinómios
- Programação matemática
  - Programação linear
- Grafos
  - Caminho mais curto…
- Contagem e enumeração
  - Contagem de estruturas combinatórias
- Computação paralela e distribuída
  - Evitar deadlock, consenso distribuído
- …

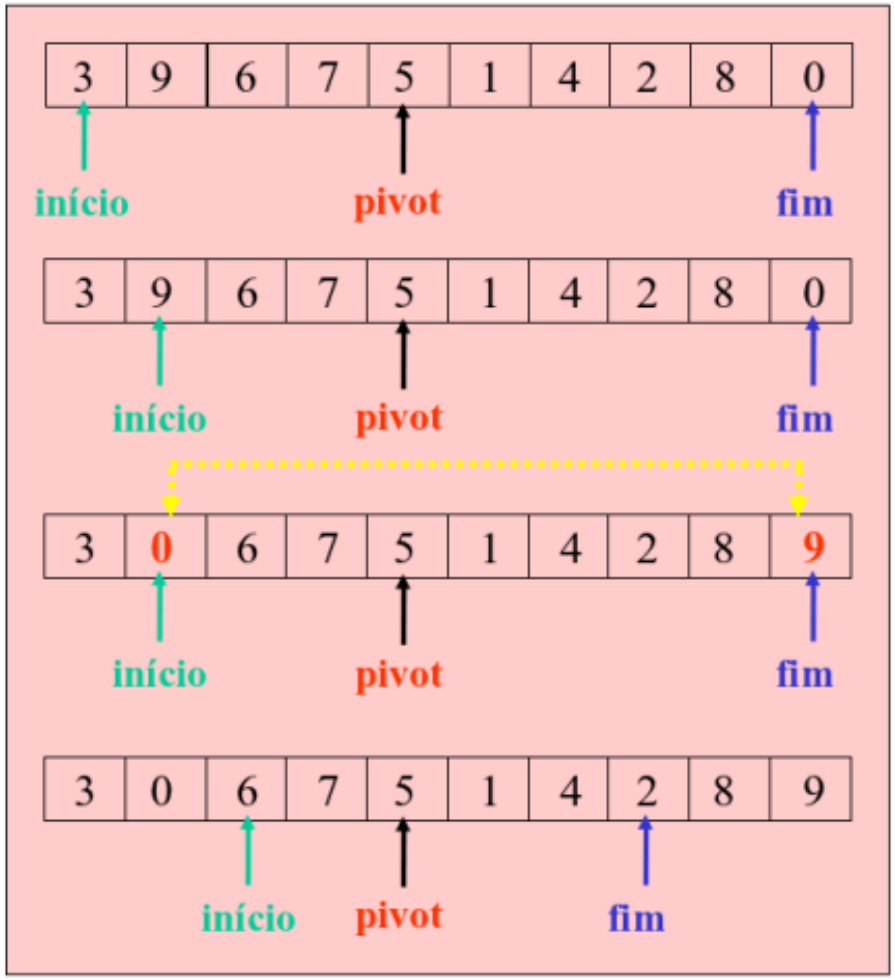
Quicksort

- O algoritmo **Quicksort** é um método de ordenação muito rápido e eficiente, inventado por C.A.R. Hoare em 1960
-
  - Quando visitou a Universidade de Moscovo como estudante,
  - Para traduzir um dicionário de inglês para russo, ordenando as palavras,
  - O objetivo era reduzir o problema original em subproblemas que possam ser resolvidos mais fácil e rapidamente.

Algoritmo

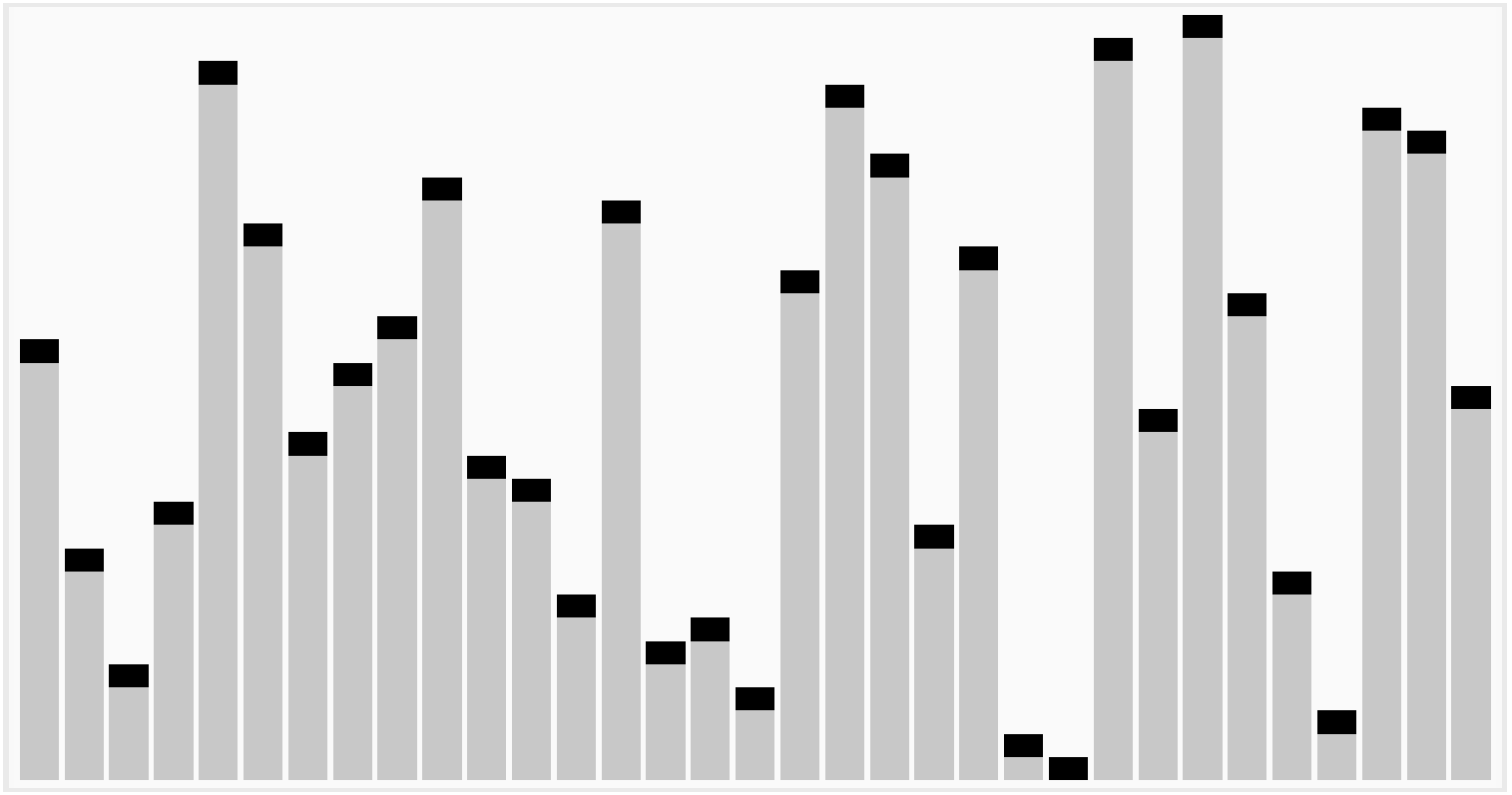
- Escolha um elemento da lista, denominado pivô;
- Rearranje a lista de forma que todos os elementos anteriores ao pivô sejam menores que ele, e todos os elementos posteriores ao pivô sejam maiores que ele.
  - Ao fim do processo o pivô estará em sua posição final e haverá duas sublistas não ordenadas.
  - Essa operação é denominada partição
- Recursivamente, ordene a sublista dos elementos menores e a sublista dos elementos maiores;
  - O processo é finito

Partição



1. Escolher o pivot;
2. Movimentar o "inicio" até encontrar um elemento maior que o pivot;
3. Movimentar o "fim" até encontrar um elemento menor que o pivot;
4. Trocar o elemento encontrado no ponto 2 com o elemento encontrado no ponto 3;
5. Recomeçar o processo (i.e. voltar ao ponto 2) até que: "inicio" > "fim"

Quick Sort em acção



Código (Java)

- De P2

```
static void quickSort(int[] a, int start, int end) {
  assert validSubarray(a, start, end);
  int n = end-start;
  if (n < 2) // should be higher (10)!
    sequentialSort(a, start, end);
  else {
    int posPivot = partition(a, start, end);
    quickSort(a, start, posPivot);
    if (posPivot+1 < end)
      quickSort(a, posPivot+1, end);
  }
  assert isSorted(a, start, end);
}

static int partition(int[] a, int start, int end) {
  int pivot = a[end-1];
  int i1 = start-1;
  int i2 = end-1;
  while(i1 < i2) {
    do
      i1++;
    while(a[i1] < pivot);
    do
      i2--;
    while(i2 > start && a[i2] > pivot);
    if (i1 < i2)
      swap(a, i1, i2);
  }
  swap(a, i1, end-1);
  return i1;
}
```

Parte do código- Partição

```
static int partition(int[] a, int start, int end) {
  int pivot = a[end-1];
  int i1 = start-1;
  int i2 = end-1;
  while(i1 < i2) {
    do
      i1++;
    while(a[i1] < pivot);
    do
      i2--;
    while(i2 > start && a[i2] > pivot);
    if (i1 < i2)
      swap(a, i1, i2);
  }
  swap(a, i1, end-1);
  return i1;
}
```

Partição com pivot aleatório

```
int partitionRandomPivot(int[] a, int start, int end) {

   //  pivot part
   int randPosition= ((int) Math.floor(Math.random()*(end-start)))+start;
   System.out.printf("Pivot will be %d\n",a[randPosition]);

   swap(a,randPosition, end-1); // new :  save pivot at last position

   //  code below is the same
   int pivot=a[end-1];
   int i1 = start-1;
   int i2 = end-1;

   while(i1 < i2) {
             //  enquanto menor que pivot
             do
             i1++;
             while(a[i1] < pivot);
             //  enquanto maior que pivot e ...
              …

             if (i1 < i2){
               swap(a, i1, i2);
             }
   swap(a, i1, end-1);  //  restore pivot
   return i1;  //  <---
}
```

Análise probabilística de algoritmos

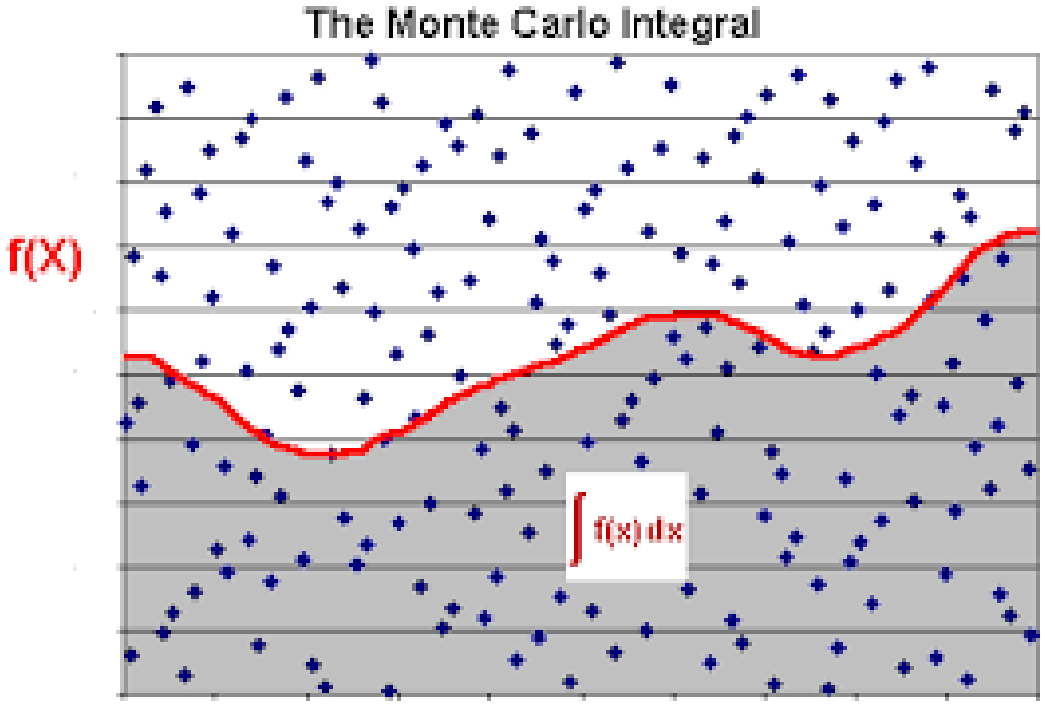
- Usa teoria de probabilidades para analisar o comportamento / desempenho de algoritmos (probabilísticos e determinísticos

- Porquê ?
  - Naturalmente, algoritmos probabilísticos terão desempenho não determinístico
  - Também, o comportamento dos alg. determinísticos varia com as entradas
  - A análise probabilística permite estimar limites

- Exemplo:
  - Determinar a probabilidade de colisão de uma função de hash (utilizada, por exemplo, em arrays associativos)

Exemplo Método Monte Carlo

- Aplicação: cálculo de valor de um integral



Probabilistic Programming

- **Probabilistic programming** is an emerging field that draws on probability theory, programming languages, and systems programming to provide concise, expressive languages for modeling and general-purpose inference engines that both humans and machines can use.

- Example of project: The MIT Probabilistic Computing Project aims to build software and hardware systems that augment human and machine intelligence.

- Picture, a probabilistic language being developed in collaboration with Microsoft, lets users solve hard computer vision problems such as inferring 3D models of faces, human bodies and novel generic objects from single images by writing short (<50 line) computer graphics programs that generate and render random scenes.

- Video of Vikash's MIT Media Lab talk on Probabilistic Programming for Augmented Intelligence
  - https://www.media.mit.edu/video/view/mansinghka-2016-03-15

Word predictor

- Filme Youtube:
  - https://youtu.be/5Mp_10tPcCU

Exemplo de simulação – Rede ALOHA

- Simulação da camada de acesso ao meio da slotted ALOHA

  ALOHA, was a pioneering computer networking system developed at the University of Hawaii. ALOHAnet became operational in June, 1971, providing the first public demonstration of a wireless packet data network. ALOHA originally stood for Additive Links On-line Hawaii Area

- **Matlab** Source at https://github.com/afcuttin/aloha

- saloha.m utiliza funções que geram números aleatórios: rand(), randi()

- Experimentem

Mais exemplos de aplicação …

- Filtrar emails com SPAM
- Máquinas de estados probabilísticas
- Parsers para análise sintáctica
- Information Retrieval
- Reconhecimento de padrões
- Reconhecimento de fala [Interacção Multimodal]
- Inteligência Artificial
  - Ex: planeamento nos robôs de Futebol robótico

Probabilidades

Conceitos essenciais

Probabilidade

- Qual a hipótese de chover amanhã?
- Qual a possibilidade de eu chegar a horas à aula ?
- Qual a probabilidade de eu ganhar o Euromilhões (ou um de vocês) ?

- São questões que colocamos frequentemente…
- e estão relacionadas com o **incerto / não determinístico**

Aleatório

- Em termos qualitativos, “qualquer coisa” que não seja predizível com certeza absoluta

- Acontecimento (evento) cujo resultado não possa ser determinado com certeza absoluta.
  - Caso contrário é determinístico

- adj. Que repousa sobre um acontecimento **incerto**, fortuito: contrato aleatório.
  Diz-se de uma grandeza que pode tomar certo número de valores, a cada um dos quais está ligada uma probabilidade.
    - De: **dicionário online de português**

- http://www.priberam.pt/dlpo/aleat%C3%B3rio

Então qual o interesse ?

- Qual o interesse em estudar algo que não se pode prever ?

- Na maioria das aplicações existe algum tipo de regularidade que se manifesta se o número de observações / experiências for elevado

Problema Exemplo 1

- Qual a probabilidade de acertar num PIN/**password** de 4 dígitos escolhendo um PIN completamente ao acaso?

  E de 20 dígitos ?

Problema Exemplo 2

- Consideremos uma família com 2 filhos

- Supondo que a probabilidade de nascer um rapaz é a mesma de nascer uma rapariga, qual a probabilidade de uma destas famílias com 2 filhos ter pelo menos 1 rapaz ?

  – De “O Acaso”, página 23

Problema Exemplo 3

- Em qual destas apostas se tem mais possibilidades de ganhar:
  - "pelo menos um seis em 4 lançamentos de 1 dado"
  - "pelo menos um DUPLO 6 em 24 lançamentos de 2 dados"

- Problema Clássico
- Problema do Cavaleiro de Méré
  - Antoine Gombaud, denominado Chevalier de Méré (séc. XVII), foi um nobre e jogador francês que se dedicou ao cálculo matemático de jogos de azar

Problema Exemplo 4

- Qual a probabilidade de 80% ou mais dos alunos de MPEI deste ano letivo que não reprovem por por faltas ter sucesso na UC?

- **O que precisamos saber**

  - Quantos alunos considerar?
  - Qual a probabilidade de cada um passar ?
    - Talvez simplificar ?

  - O que é isso de probabilidade ?

Probabilidade

- Noção de probabilidade

  "Medida do grau de certeza associado a um resultado proveniente de um fenómeno de acaso"

- Palavra usada pela primeira vez por Bernoulli (1654-1705)

Recordar …

- Experiência aleatória
  - Procedimento que deve produzir um resultado
  - Mas mesmo que seja repetido nas mesmas condições não garante que o resultado seja idêntico

- Experiência aleatória é especificada por
  - Espaço amostral
  - Conjunto de eventos
  - Lei de probabilidade

Exemplos de experiências aleatórias

- Escolher uma letra do alfabeto
- Escolher um aluno para dar exemplos de experiências aleatórias
- Lançar uma moeda 5 vezes e registar o número de caras
- Escolher um número real entre 0 e 1
- Medir e registar o intervalo de tempo entre 2 mensagens que chegam a um servidor de email

Espaço de amostragem

- Conjunto (S) de todos os resultados possíveis de uma experiência aleatória
  - Em geral representado por S (de Sample Space)
- Resultados têm de ser mutuamente exclusivos e não divisíveis

- S é discreto se for contável
  - i.e. se contiver um número finito de elementos ou se contiver um número infinito em que se pode estabelecer uma correspondência biunívoca com o conjunto dos inteiros
- S é contínuo se não for contável

- Elementos de S são designados por resultados

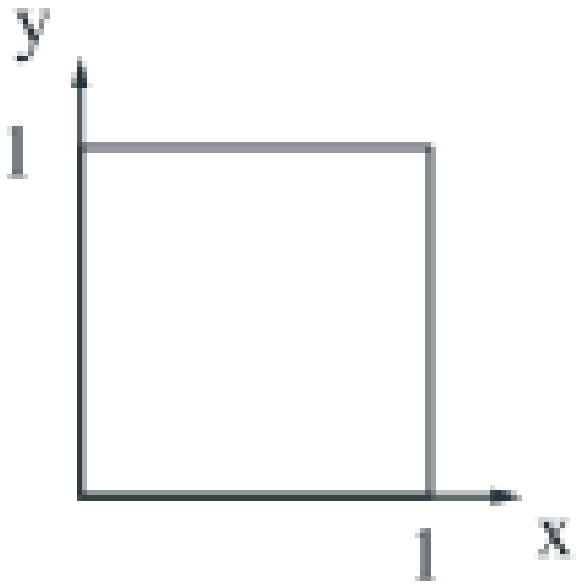
TPC

- Qual o espaço de amostragem de cada um dos problemas exemplo ?

Exemplo de espaço de amostragem contínuo

- Experiência aleatória: instante de chegada, em horas, de 2 alunos (x e y) a uma aula de 1 hora

- S= {(x,y) : 0<= x,y <=1}



Acontecimentos / eventos

- Os resultados das experiências não constituem necessariamente os únicos itens de interesse nos experimentos
  - Exemplo:
    - No caso da contagem de mensagens de email podemos estar interessados no facto de o número total exceder um determinado limiar (nº > L)
  - Os itens de interesse são representados por subconjuntos de S

- **Acontecimento (evento)** A é um subconjunto de S
  - S é obviamente um subconjunto de s próprio e constitui o evento certo
  - O conjunto vazio, ϕ, também é subconjunto, o evento impossível

- A probabilidade é atribuída a eventos

Lei de probabilidade

- Regra que atribui probabilidade aos vários eventos

- Probabilidade: número associado a um evento que indica a “verosimilhança” de esse evento ocorrer quando se efectua o experimento
  - valor entre 0 e 1
    - 1 para acontecimento certo
    - 0 para acontecimento impossível

Cálculo de probabilidades

Como é que se definem/obtêm as probabilidades associadas a eventos ?

- Através de medição
- Através da construção de modelos probabilísticos

Como determinar a probabilidade?

- Probabilidades teóricas
- **Probabilidade empíricas**

- Probabilidades subjectivas
  - Exemplo:
    - Um Médico diz que tem 95 % de certeza de que determinada pessoa tem uma determinada doença
    - Uma casa de apostas estimou em 1/5 a probabilidade de Portugal ser campeão Europeu em 2016
      - E fomos Campeões ☺
  - Não nos interessam nesta UC

Diferentes abordagens

- Teoria clássica (de Laplace)
  - Probabilidades teóricas
- Frequencista
  - Probabilidades empíricas
- Teoria matemática

Noção clássica

Simon de Laplace (1749-1827)

- "*Pour étudiér un phénoméne, il faut réduire tous les evénements du même type à un certain nombre de cas également possibles, et alors la probabilité d'un événement donné est une fraction, dont le numérateur représente le nombre de cas favorables à l'événement e dont le dénominateur représente par contre le nombre des cas possibles*"
  - pg 17 livro "O Acaso"

- Primeiro reduzir o fenómeno a um conjunto de resultados elementares, **"casos", igualmente prováveis**

$$P(evento) = \frac{número\ de\ casos\ favoráveis}{número\ de\ casos\ possíveis}$$

Exemplo

- Lançamento de 1 DADO
  - Honesto
    - => qualquer face igualmente provável
- Probabilidade de obter certa face, ex: a 5 ?
- 6 resultados ou eventos elementares
  - Representáveis pelo conjunto {1,2,3,4,5,6}
- Ao evento “saída da face 5” apenas corresponde um caso favorável
  - > P(“face 5”)=1/6

Variante do problema

- E se 2 faces tivessem o 5 marcado ?

- Espaço de amostragem ?
  - S={1,2,3,4,5} ? => casos possíveis =5
  - S={1,2,3,4,**5**,**5**}

- P(“sair 5”)=2/6

Exemplo de aplicação (em Java)

- Probabilidade de termos “0123” numa sequência de 4 dígitos
- Como fazer ? Sugestões ?
- Relembro que precisamos contar todos os casos possíveis

Resolvendo…

- Ideia 1 : 4 ciclos for …
  - Limitação do código… se quisermos 5 etc
- Ideia 2 …
  - Usar recursividade …

Possível solução

```
void comb(String example, String alphabet, int len, List<String> list)
{

    if (len == 0) {  // new combination available, store it ...
         list.add(example);  //  store it in a list
        return;
    }
    else {
        for (int i=0;i<alphabet.length(); i++){ // all alphabet

                //  recursive cal
                comb(example+alphabet.charAt(i), alphabet, len-1,list);

         }
    }
}
```

Exemplo de utilização

```
public static void main(String[] args) {

    String alphabet = "0123456789";

    final int MAX=7;

    for (int n=1;n<=MAX;n++){
       int possible=comb(alphabet,n);

       System.out.printf("t Prob=%.8f\n",1/(double)possible);
    }
```

- [PINs.java]

Regras básicas (OU)

- P("sair face maior que 4") ?

  = P("sair face 5 ou face 6") = P({5,6}) = 2/6

  = P({5})+P({6})

- P("face par")=P({2})+P({4})+P({6})=1/2
- P("qualquer face") = 6 x 1/6 =1

  … P(A U B) = P(A) + P(B)

  Sempre ???

Regras básicas

- P("face menor ou igual a 4")

  =1 – P("face maior que 4")

  = 1 – 2/6 = 4/6

Regra do complemento

$P(\bar{A}) = 1 - P(A)$

Regras básicas (E)

- P(“face par E face menor ou igual a 4”)=

  = P(“face par”) x P(“face menor ou igual a 4”)

  $= \frac{1}{2} \times \frac{2}{3} = \frac{1}{3}$

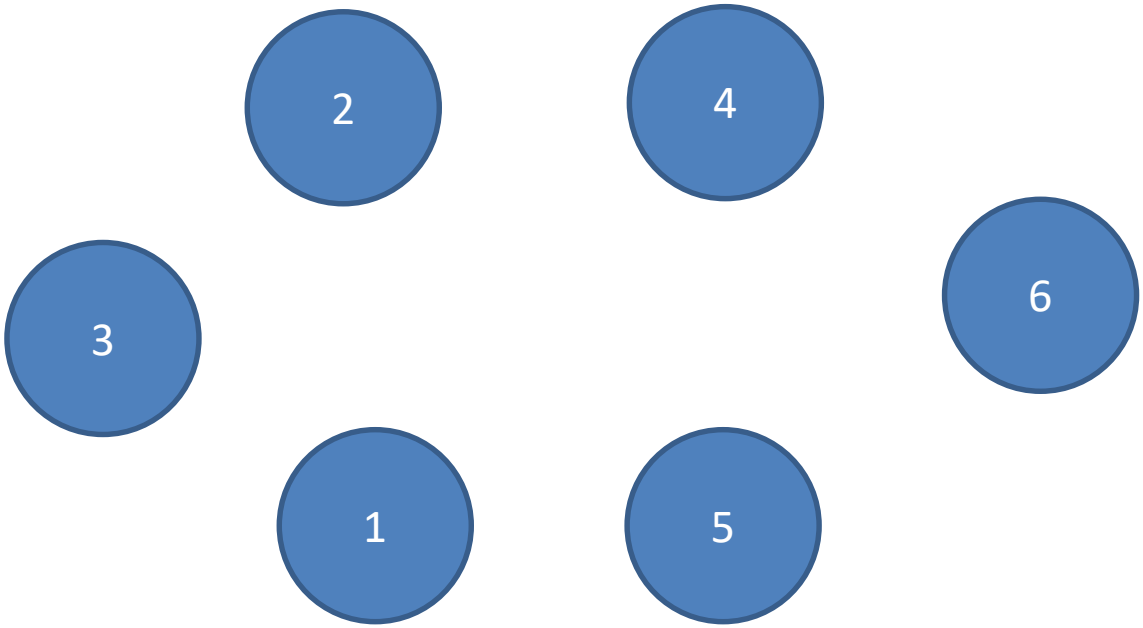
  De facto existem 2 possibilidades em 6 , {2,4}

Aplicação das regras (OU novamente)

- P(“face par OU face menor ou igual a 4”) = ?
- Se fizermos P(“face par”)+ P(“face menos ou igual a 4”) dá 7/6 > 1 !!
- Qual o erro ?

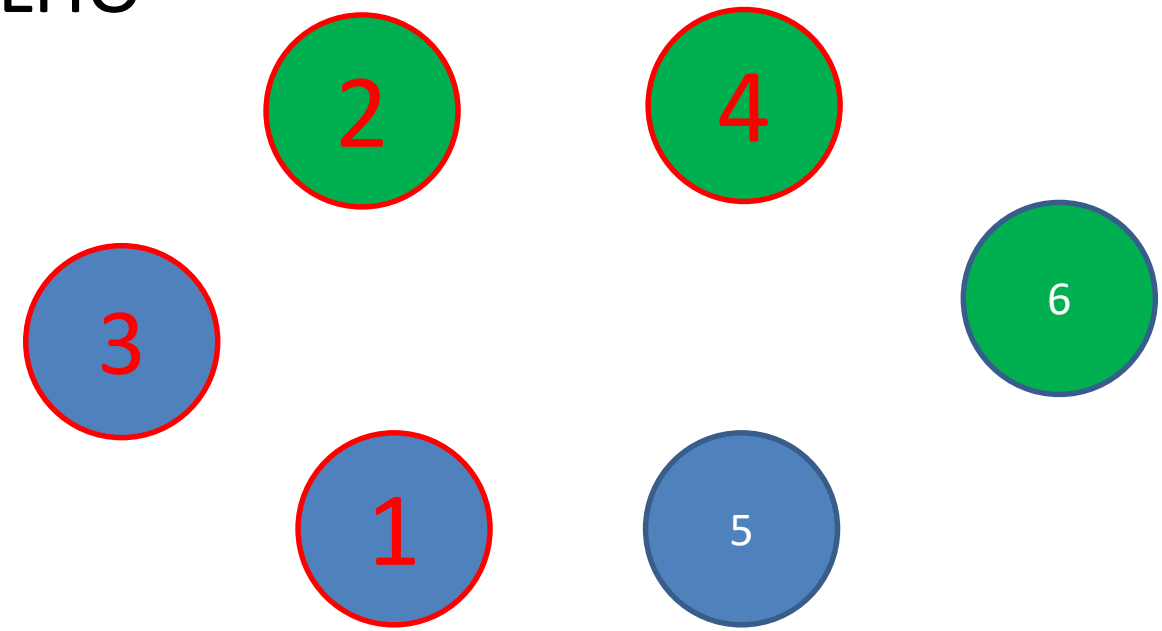
Analisemos

S



Eventos

- A=“face par” VERDE
- B=“face menor ou igual a 4” limite e texto a VERMELHO



…

Temos 3 com fundo verde => P(A) = ½

Temos 4 com vermelho => P(B) = 2/3

… mas temos 2 casos com verde e vermelho

- No mínimo perigoso ☺

- Estávamos a contar 2 vezes a intersecção

- P(A U B)= P(A)+P(B) – P(A ∩ B)
  = ½ + 2/3 - 2/6 = 3/6+4/6-2/6 =5/6

Testar as regras (num problema exemplo)

- Considere uma família com 2 filhos e que a probabilidade de nascer rapaz é igual à de nascer uma rapariga.
- Designando o nascimento de um filho por M e uma filha por F, qual a probabilidade de MF ?

- Probabilidade de pelo menos 1 rapaz numa família com 2 filhos ?

Resolução

- Pelo menos 1 rapaz => MF ou FM ou MM
- MF é a intersecção (“e”) de M no primeiro e F no segundo = ½ * ½
- Similar para MM e FM
- P(MF) + P(MM) + P(FM) = ¾
  - Devido à união (“ou”)

Problema do Cavaleiro de Méré

- Aplicação de teoria Clássica

- Criar lista de todas as possibilidades (S)
  - 4 lançamentos
    - 1111
    - 1112
    - 1113 …
  - 24 lançamentos …
- Contar casos favoráveis
- Calcular probabilidade

- Sugestão de TPC (Java?)

Problema do Cavaleiro de Méré

- P(“sair pelo menos um seis em 4 lançamentos de 1 dado”) vs P (“sair DUPLO 6 em 24 lançamentos de 2 dados”)

- Melhor usar a regra do complemento..
- P(“nenhum 6 em 4 lançamentos ”)=
- P(“não 6 na primeira E não 6 na segunda E …”)

  =P(“não 6 na primeira”) x P(“não 6 na segunda”)

…

= 5/6 x 5/6 …. = (5/6) ^ 4

- P(“sair pelo menos um seis em 4 lançamentos de 1 dado”)

  $= 1 - \left(\frac{5}{6}\right)^4$

  = 0,51775

- P (“sair DUPLO 6 em 24 lançamentos de 2 dados”) =

  $= 1 - \left(\frac{35}{36}\right)^{24}$

  = 0,49141

Não esquecer

- Estas regras e definição clássica ASSUMEM dados honestos, moedas honestas, igual probabilidade de nascer rapaz e rapariga, equiprobabilidade para os eventos elementares

- Uma questão que surge naturalmente é se na prática tais valores são ou não razoáveis ?