

Bloom Supermarket

Departamento de Eletrónica, Telecomunicações e
Informática

Métodos Probabilísticos para Engenharia Informática

Tiago Dias 88896, Bruno Lopes 89179

11 de Dezembro de 2018



universidade
de aveiro

Conteúdo

Objetivos

Introdução

Parte 1

Desenvolvimento dos módulos

- 1.1. Bloom Filter
- 1.2. Counting Bloom Filter
- 1.3. Minhash
- 1.4. Contador Estocástico

Parte 2

Teste dos módulos

- 2.1 Teste Bloom Filter
- 2.2 Teste Counting Bloom Filter
- 2.3 Teste Minhash

Parte 3

Demonstração do programa em conjunto

Conclusão

Correr o programa

Objetivos

O objetivo deste trabalho consiste na elaboração de um programa que tem como finalidade demonstrar a utilização conjunta de algoritmos probabilísticos para a verificação de pertença a um conjunto e , também, posteriormente comparação dos mesmos de modo a determinar e verificar a existência de itens similares entre conjuntos.

Introdução

Este trabalho está dividido em três partes:

1. Desenvolvimento dos módulos em cima referidos.
2. Teste dos módulos.
3. Demonstração do programa em conjunto

Bloom Supermarket: É uma plataforma de compras virtual, ou seja, é um supermercado virtual.

Para se poder usufruir do programa foi criada uma interface muito simples de utilizar, o *Login*. Caso o utilizador, que esteja a fazer *login*, não possua conta, será questionado sobre o querer criar conta, conta esta que ficará guardada para futuros acessos.

Funcionalidades:

1. Ver produtos do supermercado(por constituintes do mesmo ou através dos favoritos do utilizador)
2. Comprar produtos
3. Adicionar produtos aos favoritos

Ao fim de ser feito o *login* ou o registo existe, como opção, a área de administrador, cuja apenas pode ser acedida por administradores previamente adicionados(ex: professor da prática).

Parte 1

Desenvolvimento dos módulos

1.1. *Bloom Filter*

O primeiro módulo é o *Bloom Filter*. Este é uma estrutura de dados probabilísticos que nos informa se determinado elemento pertence a um dado conjunto ou se o elemento está no conjunto.

Os métodos pertencentes a estes módulos são os seguintes:

- **Bloom Filter(int n)**: Recebe um argumento, n(inteiro), que vai definir o tamanho do filtro. Inicializa o filtro com todas as posições a 0(vazias). Calcula, também, o número de *hash functions* que vão ser utilizadas e o número de membros que irão ser inseridos no filtro.
- **insert(E value)**: Recebe um argumento de qualquer tipo, pois o tipo E é geral, e insere-o no filtro.
- **isMember(E value)**: Recebe um argumento, outra vez geral, e verifica se este pertence ao filtro/conjunto(método lecionado nas aulas práticas e denominado com *set membership*).
- **size()**: Retorna o número de elementos do filtro.
- **toString()**: Através de *prints* no terminal são mostrados muitos constituintes do *Bloom Filter*.

1.2. *Count Bloom Filter*

Este módulo irá ter os mesmos métodos que o *Bloom Filter*, mas alguns destes serão modificados, mais dois novos denominados *remove* e *count*.

O método do *insert* irá ser modificado de modo a, neste momento, incrementar o valor dos *buckets* do filtro. O método do *isMember* verifica, neste caso, se cada um dos *buckets* necessários é diferente de zero.

O método *remove*, que recebe um valor geral, consiste em remover o valor recebido do respetivo *bucket* e decrementar o número de valores no filtro.

O método *count* consiste em calcular o conjunto de contadores e retornar o valor mínimo com uma frequência estimativa.

1.3. Minhash

Este módulo tem como ideia reduzir a variância comparando, em média, várias variáveis criadas da mesma maneira. A estimativa será igual à fração de *hash functions* que concordam.

De modo a conseguirmos apresentar uma lista de produtos por constituinte e sugerir produtos semelhantes aos favoritos de um dado utilizador, precisamos de encontrar aqueles que correspondem à pesquisa do utilizador. Para isso utilizamos distâncias de *Jaccard*. A similaridade de *Jaccard* é a divisão entre a interseção e união de dois conjuntos. Se a 1 subtrairmos esta similaridade, obtemos a distância de *Jaccard*.

1.4. Contador estocástico

Este módulo é um contador cuja probabilidade de incrementar o seu valor é cada vez menor à medida que o mesmo aumenta. Quando o contador contém n valores, a probabilidade de incrementar é de 2^{-n} .

Parte 2

Testes dos módulos

2.1. Teste do *Bloom Filter*

Preparamos dois testes para o *Bloom Filter*, ambos à base de comparação de *Strings*. O primeiro exemplo é um filtro que contém cidades de Portugal e testa se uma dada cidade pertence a esse conjunto. O segundo exemplo é um módulo que através de alguns excertos de um livro (“Os Maias”) nos pede para escrever uma palavra para ver se essa mesma está nos excertos.

Primeiro teste:

```
Modulo de teste de BloomFilter - Capitais de Distrito de Portugal Continental

Insira um nome de uma cidade para saber se é capital de um distrito de Portugal
Cidade ("fim" para acabar teste) : Porto
Porto é uma capital de distrito!
Cidade ("fim" para acabar teste) : Braga
Braga é uma capital de distrito!
Cidade ("fim" para acabar teste) : Leiria
Leiria é uma capital de distrito!
Cidade ("fim" para acabar teste) : Santarém
Santarém é uma capital de distrito!
Cidade ("fim" para acabar teste) : Portalegre
Portalegre é uma capital de distrito!
```

Segundo teste:

Modulo de teste de BloomFilter - Os Maias

```
Foram extraídos vários excertos do livro "Os Maias" de Eça de Queirós
O texto tem um total de 1991 palavras e está escrito em minúsculas
Tente adivinhar uma palavra que pertença aos excertos retirados
Palavra ("fim" para acabar teste): Sei
Sei não pertence aos excertos!
Palavra ("fim" para acabar teste): ter
ter pertence aos excertos!
Palavra ("fim" para acabar teste): casa
casa pertence aos excertos!
Palavra ("fim" para acabar teste): maia
maia pertence aos excertos!
Palavra ("fim" para acabar teste): carlos
carlos pertence aos excertos!
Palavra ("fim" para acabar teste): filipe
filipe não pertence aos excertos!
Palavra ("fim" para acabar teste): ana
ana não pertence aos excertos!
Palavra ("fim" para acabar teste): lisboa
lisboa pertence aos excertos!
```

2.2. Teste do *Counting Bloom Filter*

Para o teste do *Counting Bloom Filter*, foram utilizados, novamente os excertos de “Os Maias” e foi verificado e imprimido o número de vezes que cada palavra aparece ao longo do mesmo.

Modulo de teste de CountFilter - Os Maias

Irá ser imprimido o número total de vezes que cada palavra aparece ao longo dos excertos do livro "Os Maias"

```
pedro - 16
e - 58
maria - 19
no - 12
entanto - 15
numa - 2
felicidade - 2
de - 95
novela - 2
iam - 1
descendo - 2
a - 73
itália - 4
pequenas - 1
jornadas - 1
cidade - 2
em - 21
nessa - 1
via - 2
sagrada - 1
que - 27
vai - 2
desde - 9
as - 16
flores - 16
das - 8
meses - 1
```

É preciso ter em atenção que nesta imagem não estão presentes todas as palavras presentes nos excertos, apenas uma pequena parte das mesmas.

2.3. Teste do *Minhash*

No teste a este módulo da *Minhash*, utilizámos a *class* do Produto. Foram criados, aleatoriamente, 5 produtos e estes foram comparados através de *Minhashing*(100 Hashes).

```
Modulo de teste de MinHashing - Produtos
Produto introduzido!!Papaia - Calorias (764g) - Proteínas (2g) - Açúcar (155g) - Lípidos Saturados (2g) - Lípidos Monoinsaturados (1g) - Lípidos Poliinsaturados (2g)
Produto introduzido!!Papaia - Calorias (770g) - Proteínas (2g) - Açúcar (150g) - Lípidos Saturados (2g) - Lípidos Monoinsaturados (3g) - Lípidos Poliinsaturados (2g)
Produto introduzido!!Cabrito - Calorias (6969g) - Proteínas (1235g) - Açúcar (0g) - Lípidos Saturados (39g) - Lípidos Monoinsaturados (61g) - Lípidos Poliinsaturados
Produto introduzido!!Nabo - Calorias (263g) - Proteínas (3g) - Açúcar (31g) - Lípidos Saturados (2g) - Lípidos Monoinsaturados (1g) - Lípidos Poliinsaturados (3g) - C
Produto introduzido!!Bijou - Calorias (2311g) - Proteínas (84g) - Açúcar (75g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (3g) - Lípidos Poliinsaturados (3g)

Distancia entre Papaia e Papaia: 0$58
Distancia entre Papaia e Cabrito: 0$89
Distancia entre Papaia e Nabo: 0$65
Distancia entre Papaia e Bijou: 0$67
Distancia entre Papaia e Cabrito: 0$83
Distancia entre Papaia e Nabo: 0$69
Distancia entre Papaia e Bijou: 0$64
Distancia entre Cabrito e Nabo: 0$86
Distancia entre Cabrito e Bijou: 0$88
Distancia entre Nabo e Bijou: 0$65

Este modulo demonstra a criação de 5 produtos aleatoriamente e a sua comparação através de minHashing (100 Hashes)
```

Mais uma vez, nesta imagem nem tudo aparece. Os constituintes dos produtos continuam em grande quantidade para a direita.

Parte 3

Demonstração do programa em conjunto

1. Primeiro o utilizador efetua o *login* ou regista-se e de seguida entrará no menu principal:

```
-----
| Bloom Supermarket |
-----
| 1 - Login          |
| 2 - Registar       |
| 0 - Sair           |
-----

Que deseja fazer: 1
Email: bruno25@ua.pt
Password: 12346
Login confirmado.
```

Bloom Supermarket

Logged In as: bruno25@ua.pt

- ```
|1 - Comprar
|2 - Ver Produtos
|3 - Listar Produtos Favoritos
|4 - Adicionar Produto aos Favoritos
|5 - Sugestões de Produtos
|6 - Area de Admin
|0 - Sair
```

Que deseja fazer:

Que deseja fazer: 2

- 1 - Sardinha - Calorias (15g) - Proteínas (3g) - Açúcar (0g) - Lípidos Saturados (2g) - Lípidos Monoinsaturados (
- 2 - Bijou - Calorias (2321g) - Proteínas (84g) - Açúcar (75g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados
- 3 - Cabrito - Calorias (6971g) - Proteínas (1232g) - Açúcar (0g) - Lípidos Saturados (37g) - Lípidos Monoinsatura
- 4 - Nabo - Calorias (256g) - Proteínas (3g) - Açúcar (31g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (2g
- 5 - Papaia - Calorias (762g) - Proteínas (2g) - Açúcar (155g) - Lípidos Saturados (2g) - Lípidos Monoinsaturados
- 6 - Sal - Calorias (0g) - Proteínas (0g) - Açúcar (0g) - Lípidos Saturados (0g) - Lípidos Monoinsaturados (0g) -

Que deseja fazer: 1  
Disponibilidade: 2 - 1 - Sal - Calorias (0g) - Proteínas (0g) - Açúcar (0g) - Lípidos Saturados (0g) - Lípidos Monoinsaturados (0g) - Lípidos Poliinsaturados (0g)  
Disponibilidade: 2 - 2 - Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (3g) - Lípidos Poliinsaturados (3g)  
Disponibilidade: 2 - 3 - Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (1g) - Lípidos Poliinsaturados (1g)  
Disponibilidade: 2 - 4 - Sardinha - Calorias (16g) - Proteínas (2g) - Açúcar (0g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (1g) - Lípidos Poliinsaturados (1g)  
Voltar atrás - 0  
Escolha o produto que quer comprar: 3

Que deseja fazer: 3

Sal - Calorias (0g) - Proteínas (0g) - Açúcar (0g) - Lípidos Saturados (0g) - Lípidos Monoinsaturados (0g) - Lípidos Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Lip Papia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) - Sardinha - Calorias (16g) - Proteínas (2g) - Açúcar (0g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - L



5. Adicionar Produto aos Favoritos aqui é pedido ao utilizador o nome do produto, pode saber o nome de um novo na opção de ver produtos:

Que deseja fazer: 4

Insira o nome do Produto: Cabrito

Produto adicionado aos favoritos

Voltando a ver a lista de produtos favoritos verificamos que foi, com sucesso, adicionado cabrito à mesma.

Que deseja fazer: 3

Sal - Calorias (0g) - Proteínas (0g) - Açúcar (0g) - Lípidos Saturados (0g) - Lípidos Monoinsaturados (0g) - Lípidos Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Líp Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) - Sardinha - Calorias (16g) - Proteínas (2g) - Açúcar (0g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - L Cabrito - Calorias (6994g) - Proteínas (1213g) - Açúcar (0g) - Lípidos Saturados (37g) - Lípidos Monoinsaturados (62

6. Sugestão de Produtos (esta operação será um conjunto de produtos semelhantes aos seus favoritos):

Que deseja fazer: 5

Porque gosta de Sal -> Calorias: 0, Proteínas: 0, Açúcar: 0, Lipidos Saturados: 0, Lípidos Monoinsaturados: 0, Lípid Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) - Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Líp Cabrito - Calorias (6994g) - Proteínas (1213g) - Açúcar (0g) - Lípidos Saturados (37g) - Lípidos Monoinsaturados (62

Porque gosta de Nabo -> Calorias: 258, Proteínas: 2, Açúcar: 30, Lipidos Saturados: 3, Lípidos Monoinsaturados: 1, L Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) - Sardinha - Calorias (16g) - Proteínas (2g) - Açúcar (0g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - L Cabrito - Calorias (6994g) - Proteínas (1213g) - Açúcar (0g) - Lípidos Saturados (37g) - Lípidos Monoinsaturados (62

Porque gosta de Papaia -> Calorias: 760, Proteínas: 5, Açúcar: 158, Lipidos Saturados: 1, Lípidos Monoinsaturados: 3 Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Líp Sardinha - Calorias (16g) - Proteínas (2g) - Açúcar (0g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - L Cabrito - Calorias (6994g) - Proteínas (1213g) - Açúcar (0g) - Lípidos Saturados (37g) - Lípidos Monoinsaturados (62

Porque gosta de Sardinha -> Calorias: 16, Proteínas: 2, Açúcar: 0, Lipidos Saturados: 1, Lípidos Monoinsaturados: 2, Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) - Bijou - Calorias (2326g) - Proteínas (84g) - Açúcar (74g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Líp

Porque gosta de Cabrito -> Calorias: 6994, Proteínas: 1213, Açúcar: 0, Lipidos Saturados: 37, Lípidos Monoinsaturado Bijou - Calorias (2326g) - Proteínas (84g) - Açúcar (74g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (2g) - Nabo - Calorias (258g) - Proteínas (2g) - Açúcar (30g) - Lípidos Saturados (3g) - Lípidos Monoinsaturados (1g) - Líp Papaia - Calorias (760g) - Proteínas (5g) - Açúcar (158g) - Lípidos Saturados (1g) - Lípidos Monoinsaturados (3g) -

7. Área de Administrador (Área restrita aos administradores da plataforma, onde estes podem gerir o inventário dos produtos, consultam os utilizadores registados e efetuam a “limpeza” da base de dados ou adicionam produtos aleatórios ao stock):

| Bloom Supermarket |

| Admin Area |

| 1 - Encher o Stock |

| 2 - Listar Users |

| 3 - Limpar Base de Dados |

| 4 - Adicionar Produtos Aleatórios |

| 0 - Sair |

Opção:

Escolhendo a opção 1 é reenchido o stock.

Na opção 2 são listados no terminal todos os utilizadores presentes no programa (alguns deles gerados aleatoriamente).

Seguidamente, na opção 3, é feito um *delete* à base de dados.

Na quarta são adicionados à lista produtos aleatórios.

## Conclusão

De acordo com os objetivos do trabalho foi desenvolvido um programa em Java que determina a pertença a um conjunto, deteta itens similares entre conjuntos e efetua uma contagem estocástica. Assim, após a realização dos testes ao programa evidenciamos que está funcional, uma vez que foi possível verificar a utilização conjunta dos três módulos.

## Correr o programa

Abrindo um IDE (de preferência com um com consola própria), abre-se um projecto a partir de um ficheiro e seleciona-se o ficheiro Projeto.zip submetido. Executa-se o ficheiro App.java do package “bloomSupermarket”, de modo a aceder à plataforma. Na área de administradores existem administradores previamente adicionados (os dois alunos, Bruno e Tiago, e o professor da prática). As contas estão no ficheiro AppData.java mas para maior eficiência deixamos aqui a conta atribuída ao professor (email: [ajst@ua.pt](mailto:ajst@ua.pt) , pass: EprecisoTrabalhar). Para testar o *Bloom Filter*, o *Counting Bloom Filter* e a *MinHash*, execute os ficheiros TesteBloomFilter1.java, TesteBloomFilter2.java, TestCountFilter.java e MHTeste.java do package “tests”.