

Padrões Climáticos

Introdução

Como nos foi proposto nas aulas de MPEI, implementámos, testámos e iremos demonstrar, uma aplicação que irá utilizar um conjunto de funções que nos permitem detetar várias sequências(BloomFilter), bem como verificar se 2 localidades têm temperaturas semelhantes(MinHash)

Descrição de como correr os vários programas

O nosso projeto, com base no tema de “Padrões Climáticos”, cria um array de temperaturas relativas a um ano de várias Localidades. Este array contém temperaturas hora a hora de 31 dias relativos a 12 meses.(Por padrão definimos que todos os meses têm 31 dias)

Para executar a aplicação, basta apenas executar o ficheiro “Main.java”, o qual irá invocar os métodos e funções necessárias para o seu bom funcionamento. Ao executar a Main iremos ser confrontados com 4 opções(introduzir um inteiro para selecionar). Destas 4 opções, uma utiliza a parte relacionada com o BloomFilter (opção (1)). Nesta função, podemos executar 2 tarefas diferentes: 1ª para procurar uma sequência de temperaturas à escolha do utilizador, 2ª apresentar todas as sequências de temperaturas(hora a hora) iguais que ocorrem ao longo de um dia de uma certa Localidade. Relativamente à 2ª opção da Main, esta utiliza a parte relacionada com a MinHash. Nesta mesma, pedimos ao utilizador um dia de um certo mês, de uma respetiva localidade e comparamos as temperaturas desse dia, com as temperaturas dos outros dias bem como dos dias dos outros meses. Para isso pedimos ao utilizador a similaridade de Jaccard pretendida e o programa encontrará dias com uma similaridade igual ou superior à pedida. Na 3ª opção, usamos a MinHash para comparar temperaturas(diárias, mensais ou anuais) entre duas localidades. Por último, numa 4ª opção, oferecemos a possibilidade de mostrar ao utilizador certas estatísticas(impressão de dias de um mês, impressão da média de temperaturas diárias de um mês e impressão da média mensal de um ano).



Testes

Para a realização dos testes, tivemos em conta algumas questões de eficácia relativamente ao número de funções de “Hash” que iríamos necessitar, bem como o tamanho do array para o bloomFilter. Após fazermos os cálculos que beneficiariam a redução de falsos positivos, executamos alguns testes relativos ao “BloomFilter”, e ao “MinHash”.

Para executar os testes, basta correr o ficheiro “*testBloomFilter.java*” relativo ao BloomFilter e o “*testMinhash.java*”, relativo ao Minhash.

1º Teste relativo ao BloomFilter

```
8 BloomFilter b= new BloomFilter(100000,3);
9
10 String temperaturas[] = {"12","13","14","15","16","17","18","15","15","15","14","13","12","17","18","17","17","17","13","17","18","19","12","11","14"};
11 //para sequencias de 3 temperaturas(n=3)
12 int n=3;
13 String r="";
14 for(int j=0; j<n; j++) { //cria as sequencias todas possiveis com 3 valores!
15     for(int i=0; i<temperaturas.length; i++) {
16         if(i%n==0) {
17             r+= " ";
18         }
19         if(i+j<temperaturas.length) r+=temperaturas[i+j]+"/";
20     }
21 }
22 System.out.printf("%s\n", r); //----->Imprime a string
23 String splitada[] = r.split(" ");
24
25 for(int i=0; i<splitada.length; i++){ //Inserir todas as sequencias possiveis no bloom
26     String[] s = splitada[i].split("/");
27     if(s.length == n){ //so inserimos as sequencias de tamanho pedido, pois ha alturas em que a dividir o array, se cria sequencias menores
28         b.inserir(splitada[i]);
29     }
30 }
31
32 //varios testes para ver se temperaturas pertencem
33 if(b.pertence("12/13/14/")) System.out.println("pertence");// tem de imprimir esta mensagem
34 else System.out.println("Nao pertence");
35
36 if(b.pertence("13/14/15/")) System.out.println("pertence");// tem de imprimir esta mensagem
37 else System.out.println("Nao pertence");
38
39 if(b.pertence("12/13/14/")) System.out.println("pertence");// tem de imprimir esta mensagem
40 else System.out.println("Nao pertence");
41
42 if(b.pertence("19/12/11/")) System.out.println("pertence");// tem de imprimir esta mensagem
43 else System.out.println("Nao pertence");
44
45 if(b.pertence("12/11/15/")) System.out.println("pertence");
46 else System.out.println("Nao pertence");// tem de imprimir esta mensagem
47 if(b.pertence("11/14/")) system.out.println("pertence");
48 else System.out.println("Nao pertence");// tem de imprimir esta mensagem pois apesar de se gerar a sequencia "11/14/", esta não é introduzida
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL

12/13/14/ 15/16/17/ 18/15/15/ 15/14/13/ 12/17/18/ 17/17/17/ 13/17/18/ 19/12/11/ 14/ 13/14/15/ 16/17/18/ 15/15/15/ 14/13/12/ 17/18/17/ 17/17/13/ 17/18/19/ 12/11/14/ 14/15/16/ 17/18/15/ 15/15/14/ 13/12/17/ 18/17/17/ 17/13/17/ 18/19/12/ 11/14/

pertence
pertence
pertence
Nao pertence
Nao pertence

2: Java Process Console +

Ao realizar este teste, criámos um array de Strings que iria representar a as temperaturas diárias(24 temperaturas hora a hora). Feito isto, usamos a mesma função que o nosso programa usa, para converter o nosso array de Strings, numa String com todas as combinações de sequências possíveis, obtendo uma String. Depois usamos o método split(), para obtermos um array de Strings com todas as combinações possíveis de sequências. De seguida introduzimos cada String(que corresponde a uma sequência possível) que tem o tamanho de sequencia

pretendido no BloomFilter e realizámos vários testes de “pertença”, que deram todos o resultado que esperámos.

2º Teste relativo ao MinHash

```
4 MinHash min = new MinHash(50);
5 String a="18/1/1 19/2/1 19/3/1 18/4/1 19/5/1 20/6/1 21/7/1 22/8/1 17/9/1 19/10/1 18/11/1 20/12/1 15/13/1 19/14/1 19/15/1 16/16/1 17/17/1 18/18/1 21/19/1 19/20/1 22/21/1 22/22/1 21/23/1 20/24/1 ";
6 String[] aa= a.split(" ");
7
8 String b="5/1/1 9/2/1 19/3/1 12/4/1 19/5/1 30/6/1 21/7/1 22/8/1 23/9/1 19/10/1 1/11/1 20/12/1 15/13/1 19/13/1 1/15/1 15/16/1 17/17/1 12/18/1 21/19/1 19/20/1 22/21/1 22/22/1 21/23/1 20/24/1 ";
9 String[] bb= b.split(" ");
10
11 String teste1="18/1/1 13/2/1 14/3/1 18/4/1";
12 String teste2="18/1/1 13/2/1 14/3/1 18/4/1"; //----->São iguais!!!
13 String teste1[] = teste1.split(" ");
14 String teste2[] = teste2.split(" ");
15
16 Set<String> setPedida = min.CreateProfileSet(aa); //cria um "Set" com as temperaturas(com etiquetas)
17 Set<String> setPedida1 = min.CreateProfileSet(bb);
18
19 Set<String> setTeste1 = min.CreateProfileSet(teste1); //cria um "Set" com as temperaturas(com etiquetas)
20 Set<String> setTeste2 = min.CreateProfileSet(teste2);
21
22 double disteste1 = min.disJaccard(setPedida, setPedida1); // Obter a distancia de jaccard
23 double disteste2 =min.disJaccard(setTeste1, setTeste2);
24
25 System.out.println("-----");
26 System.out.println("Primeiro teste");
27 if(disteste1>0.7){
28     System.out.println("\n\nAs temperaturas são muito semelhantes") ;
29     System.out.printf("Distancia de Jaccard %.2f\n", disteste1);
30 }
31 else{
32     System.out.println("\n\nAs temperaturas são diferentes");
33     System.out.printf("Distancia de Jaccard %.2f\n", disteste1);
34 }
35 System.out.println("-----");
36 System.out.println("Segundo teste\n");
37 if(disteste2==1){
38     System.out.println("As temperaturas de teste sao iguais"); //-----> É suposto imprimir este pois sao 100% iguais
39     System.out.printf("Distância de Jaccard: %f\n", disteste2);
40 }
41 else{
42     System.out.println("\n\nAs temperaturas sao diferentes");
43     System.out.printf("Distancia de Jaccard %.2f\n", disteste2);
44 }
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL

2: Java Process Console + - [] ^ x

Primeiro teste

As temperaturas são diferentes
Distancia de Jaccard 0,41

Segundo teste

As temperaturas de teste sao iguais
Distância de Jaccard: 1,000000

Para realizarmos este teste criámos uma String de temperaturas do tipo 23/1/1, em que o 23 se refere á temperatura do dia 1 do mês 1. Fizemos um processo idêntico ao do primeiro teste, neste caso introduzindo cada temperatura de cada Localidade (18/1/1 19/2/1 ...) num “TreeSet” dentro da função MinHash. Fizemos 2 tipos de comparações, uma com 24 temperaturas(algumas iguais e outras diferentes), obtendo um resultado positivo para a respetiva distância de Jaccard. No segundo teste, criamos 2 Strings de temperaturas iguais, obtendo uma distância de jaccard de 1, significando assim que as 2 Strings seriam iguais.

Vantagens e Limitações

Perante esta aplicação, vimos que conseguimos verificar se havia determinadas sequências num certo mês, mas também verificar todas as sequências possíveis de um

determinado dia com elevada rapidez. Outra grande vantagem foi a utilização do Minhash para verificar quais dias de um ano eram iguais(conforme a distancia de Jaccard introduzida) em relação ao dia introduzido pelo utilizador num intervalo de tempo pequeníssimo.

Uma limitação foi a ocorrência de alguns falsos positivos ao usar o BloomFilter, indicando que uma certa sequência pertencia, não devendo pertencer.