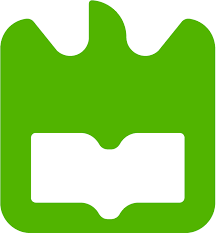
Universidade de Aveiro

Projeto de Métodos Probabilísticos para Engenharia Informática

Guilherme Henriques nº80100

João Coelho nº80335

Mestrado Integrado em Engenharia de Computadores e Telemática



**Introdução**

Com a realização deste trabalho tentámos aplicar os conhecimentos adquiridos ao longo da cadeira de Métodos Probabílisticos para Engenharia Informática relativos a BloomFilter, Distância de Jaccard e Min-Hashing. Aprendemos estes em linguagem MatLab e tivemos que os implementar em linguagem java. Achamos que este método foi ótimo para uma melhor aprendizagem pois, deste modo, tivemos que aplicar o que aprendemos noutra linguagem de programação e assim, percebermos se realmente tínhamos os devidos conhecimentos.

**Módulos Implementados**

Neste projeto decidimos implementar três módulos principais: “BloomFilter.java”, o “finder.java” e o “ua.java”. Além destes, implementámos também dois testes simples relativamente a cada um dos módulos “BloomFilter.java” e “finder.java”. Em relação ao teste conjunto, este é composto por três ficheiros/classes: “cadeira.java”, “curso.java” e “ua.java” em que a main se situa no “ua.java”.

**Utilização dos testes**

**Bloom filter:**

Para utilizar o ficheiro de testes de bloom filter, basta correr o ficheiro “BloomFilterTest.java” e seguir as instruções que aparecem no ecrã. O objetivo deste ficheiro é interagir com o utilizador de forma a que este utilize o bloomfilter como queira. Inicialmente será pedido ao utilizador que introduza a probabilidade de falsos positivos e também o número de itens no filtro. O utilizador, depois, poderá adicionar elementos ao filter ou até verificar se são membros já existentes no filter.

**Finder:**

Para utilizar o ficheiro de testes do finder, basta correr o ficheiro “finderTest.java” e seguir as instruções que aparecem no ecrã. O objetivo deste ficheiro é interagir com o utilizador de forma a que este utilize o MinHash como queira. Inicialmente será pedido ao utilizador que introduza o número de HashFunctions e o número de k shingles. O utilizador, depois, poderá comparar dois ficheiros, ver a distância entre eles e ver a similaridade destes.

**Demonstração em conjunto**

A aplicação desenvolvida serve para criar uma base de dados de cursos e das respetivas cadeiras, podendo estes ser comparados uns com os outros em termos de cadeiras e de ects para se detetar cursos que tem várias cadeiras similares através de minHash. Podemos também verificar se um curso já existe na base de dados através de bloomfilter.

**Referência a fontes utilizadas na elaboração do trabalho**

<https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>

Slides das aulas TP15 até 19

<http://matthewcasperson.blogspot.pt/2013/11/minhash-for-dummies.html>