ID2222 Lab 1 report

Fynn van Westen and Axel Myrberger

November 2021

1 The solution and instructions

The code is divided into 3 parts: algo_lib.py, preprocessor.py and run.py. algo_lib.py contains all the algoritms needed for the task including the bonuspoints, i.e point algoritm 1-5 in the task. preprocessor.py contains a simple class to load the data and clean the data. Run.py is the script that is used to to run and test the code with interaction from algo_lib.py and the preProcessor.py files. Parameters can be supplied to the runner file, while resonable default parameters are chosen already. All functions are called as part of testing the code with corresponding outputs.

The algorithms are build as outlined in the assignment instructions. One class for Shingling, CompareSets, MinHashing, CompareSignatures and LSH.

Class shingling is implemented by functions to create shingles, get shingle has functions and creating the characteristic matrix.

Class compare sets is a static method to compute the jaccard similarity:

$$\frac{set1 \cap set2}{set1 \cup set2} \tag{1}$$

Class minhasing creates signature matrices from the equation:

$$h(x) = (ax + b) mod c \tag{2}$$

This is practically implemented by using lists with random unique values, a and b. C is a large prime number implemented with nextprime method via the module sympy.

Class compareSignatures is a static method to estimate the jaccard similarity between two signatures.

Class LSH implements local sensitivity hashing. This is done by defining a number of bands when initiating the class. One function compare candidates is used to compare the estimated jaccard similarity between against a threshold, that's calculated from the numbers of bands and the rows per band. Find candidates is where the function is finding local candidates from all bands estimating similarity by bucketing them together via hashing. The combination of all possible candidates are return to be tested, in pairs of two.

To run the code install the packages from the requirements.txt and run the run.py file. Change the default values in run.py if you are interested in testing other configurations.

2 Results

The algorithms were tested on documents where the follow changes were made:

1. Original text from dataset
2. Just added one word "EXTRA" in the very beginning of the original text
3. Added some characters at random in the text
4. Added extra text of roughly 60 words in the end of the document
5. Deleted the first paragraph of the original document(shortened)
6. Complete different text from dataset
7. Version of document_5 above, with some deleted lines

Computed signatures with length 1000! The tabels row and column index indicate the document IDs, the tuple in the cell has as a first entry the exact jaccard similarity from part 1 and as second entry the estimated jaccard similarity from the signatures of part2.

1	(1.0, 1.0)	(0.996, 0.99)	(0.779, 0.766)	(0.797, 0.801)	(0.599, 0.617)	(0.001,0.002)	(0.0019,0.002)
2	(0.99,0.999)	(1.0,1.0)	(0.77,0.765)	(0.795,0.801)	(0.596,0.616)	(0.0018,0.002)	(0.0019,0.002)
3	(0.779, 0.766)	(0.777,0.765)	(1.0, 1.0)	(0.638,0.629)	(0.492,0.507)	(0.0018, 0.002)	(0.0019, 0.002)
4	(0.797, 0.801)	(0.794, 0.801)	(0.638,0.629)	(1.0, 1.0)	(0.477, 0.485)	(0.0017,0.002)	(0.0018,0.002)
5	(0.599, 0.617)	(0.597, 0.616)	(0.492, 0.507)	(0.477, 0.485)	(1.0, 1.0)	(0.0012,0.002)	(0.0013,0.002)
6	(0.0018, 0.002)	(0.0018,0.002)	(0.0017,0.002)	(0.0017,0.002)	(0.0013,0.002)	(1.0,1.0)	(0.801, 0.808)
7	(0.0019, 0.002	(0.0019,0.002)	(0.0019,0.002)	(0.00185,0.002)	(0.00135,0.002)	(0.801,0.808)	(1.0, 1.0)

Running LSH...

The Rows per band and threshold are calculated from the formulas given in the book, and are as follows. Rows per band used: 20 Threshold computed from values: 0.8223401594268891

LSH finds the following documents to be potentially similar, and the jaccard similarity confirms this.

- DOC 0 and DOC 1 have similarity 0.999
- DOC 0 and DOC 3 have similarity 0.801
- DOC 1 and DOC 3 have similarity 0.801
- DOC 5 and DOC 6 have similarity 0.808

Example Interpretation: (Keep the high threshold in mind) DOC 0 and 1 are the same except for the first word leading to high similarity. Doc 5 and 6 differ in a couple lines and are therefore similar. Notice that no document combination of (0,1,2,3,4) and (5,6) is checked by LSH. Each of these two sets has a different text as a base, from which changes were made. Therefore it makes sense that these are very different and don't get checked by LSH.