Weekplan: Nearest Neighbor and Locality-Sensitive Hashing

02807 Computational Tools for Data Science

References and Reading

[1] Chap. 3 of Mining of Massive Data Sets, Jure Leskovec, Anand Rajaraman, and Jeff Ullman.

Exercises

The exercises gradually build the components for an efficient nearest neighbor data structures on a collection of documents.

1 [w] **Setup** Download the test data and template file `similarity.py`.

2 [w] **q-grams** Implement a function `shingle` that take an integer q and a string and produces a list of shingles, where each shingle is a list of q words.

3 **Minhashing** Solve the following exercises.

3.1 Implement a minhash algorithm `minhash` that takes a list of shingles and a seed for the hash function mapping the shingles, and outputs the minhash. Feel free to use the `listhash` function in the template.

3.2 Extend the minhash algorithm to output k different minhashes in a an array. Use different seeds for each minhash, e.g., $1, \ldots, k$.

4 **Signatures** Construct a function `signatures` that takes the `docs` dictionary and outputs a new dictionary consisting of document id's as keys and signatures as values.

5 **Jaccard Similarity** Implement a function `jaccard` that takes two document names and outputs the estimated Jaccard similarity using signatures.

6 **Find Similar Items** Implement a function `similar` that finds all pairs of documents whose estimated Jaccard similarity is ≥ 0.6. Test your program for different values of k and q. Compare your results for most similar documents with your own visual impression of the similarity of files.

7 **Locality-Sensitive Hashing** Use locality-sensitive hashing to speed up your solution to the find similar item exercise.