Assignment1: Locality Sensitive Hashing

W. Kowalczyk

wojtek@liacs.nl

22.09.2014

Introduction

The purpose of this assignment is to implement the key components of the LSH method for two similarity measures: Jaccard similarity and Cosine similarity, and gain some hands-on experience with this method applied to some synthetic and real data sets.

Task 1: Implement a function jsim(s1, s2) which calculates the Jaccard similarity of sets s1 and s2. Arguments s1 and s2 are supposed to be sets (in Python's terminology), i.e., repetitions of the same element are not possible (in contrast to bags).

Task 2: Implement a function minhash(S, k) which takes as input a list of sets S and an integer k and returns two objects: a sorted list of all elements that occur in S, Words, and a matrix of size k x|S| of minhash signatures of length k, Signatures. The matrix should consist of integers - indices of elements stored in Words. Here we will assume that the elements of sets from S are either strings or integers.We use the Python convention: the integer 0 represent the first element on the list, i.e., Words[0], 1 represents the second element, etc. You don't have to implement hash functions yourself - just search Python documentation to find a suitable package. To make sure that the working of minhash(S, k) can be reproduced, add a yet another argument to it, seed, and modify it accordingly. In other words, implement minhash(S, k, seed). Consider allowing some default values on k and seed, e.g., k=10, seed=123.

Remark: There are two ways of implementing the minhash function: a) with help of random permutations of integers from range(len(Words)) - that's easy to implement, but is expensive in terms of time and memory; b) with help of 'true' hash functions, as described in Section 3.3.5 of the textbook. Implement both versions!

Task 3: Implement a function sigsim(ss1, ss2) which calculates the similarity of signatures ss1 and ss2. Note that it doesn't matter if the signatures consist of integers (as in the case of minhashing) or of -1's and +1's (as in the case of sketches) - the procedure is the same. Use the sigsim function to implement a function simmat(M) which calculates the full similarity matrix of all column vectors that are returned by the minhash functions that you developed in Task 2.

Task 4: Implement the banding technique, as described in 3.4.1-3.4.3. Demonstrate working of this method on an artificially generated set of all nonempty subsets of {1, 2, ..., 10}. More precisely:

1) Generate all non-empty subsets of {1, 2, ..., 10} and calculate the Jaccard similarity between all pairs of these sets.

2) Calculate the matrix of minhash signatures, using the number of hash functions k=100.

3) Apply your implementation of the banding technique to these signatures, using combinations of (b, r) ranging over (2,50), (5,20), (10,10), (20, 5) (50,2). Measure the performance of these 5 configurations by calculating the " empirical chance of being detected" at points s= 0.2, 0.3, ..., 0.8 and compare it to the "theoretical chance of being detected", computed with help of the formula from Section 3.4.2 (see Example 3.11). Are the empirical results close to theoretical ones?

4) You may repeat your experiment several times and average the results. Finally, plot the five theoretical "S-curves" (one for each setup).

5) Finally, create a notebook which demonstrates working of your 'LSH-Toolbox' (including your comments!). Submit the notebook via e-mail, using the subject line: AiDM_A1A. This task will be graded as "passed" or "failed".

Task 5: Implement a function cossim(s1, s2) which calculates the cosine similarity of two vectors of doubles, s1 and s2. Carefully study the definition of the cosine similarity, as discussed in the textbook or on the slides from lecture 3: it is based on the angular distance between vectors.

Task 6: Implement a function sketch(M, k) which takes as input an array of doubles M, where each column represents a point in d dimensional space (d>1), and a number of random directions, k, and returns a matrix of sketches, Sketches, which consists of column vectors of -1's and 1's of length k that represent sketches of column vectors from M. When generating k random directions simply use vectors of normally distributed random numbers and not vectors of -1 and +1, as suggested in Section 3.7.3. "Binary directions" work fine in high dimensional spaces, but we are also interested in experimenting with low dimensional spaces.

[Important to remember: the term 'sketch' is more general than suggested in Section 3.7.3: the random projections might consist of arbitrary numbers and not just of -1's and +1's!]

Task 7: Repeat similar experiments as in Task 4; this time with the cosine similarity measure. Generate 1000 random vectors in two dimensional space (in Numpy terminology: a matrix randn(2, 1000)) and calculate the cosine similarity between all pairs of these vectors. Calculate the matrix of sketches, using k=100 random projects. Then apply your implementation of the banding technique to these vectors, using the combinations of (b, r) ranging over (2,50), (5,20), (10,10), (20, 5) (50,2). Measure the performance of these 5 configurations by calculating the " empirical chance of being detected" at points s= 0.2, 0.3, ..., 0.8 and compare it to the "theoretical chance of being detected", computed with help of the formula from Section 3.4.2 (see Example 3.11). Are the empirical results close to theoretical ones? You may repeat your experiment several times and average the results. Finally, plot the five theoretical "S-curves" (one for each setup). Submit the notebook via e-mail, using the subject line: AiDM_1B.

MAIN TASK: Once your LSH-Toolbox is ready, you can apply it to a real-life problem. Look around for a suitable problem and a data set (text documents, images, strings, blocks of binary data, etc.) . Run experiments and document your findings in a report. Come up with your proposals on Monday, 29.09.2014, 9:00! Your solution (the report and a link to a repository with your data and code) should be submitted by email with the subject line AiDM_1.

Organization. You are supposed to work in couples. The deadline for this assignment is:

Monday, 13.10.2014, 23:59.