Programming and Classification Summer 2019

Text signatures and hash functions — 01.04, 2019

dr hab. inż. Marek Klonowski *Scribe: Krzysztof Agieńczuk*

1 Overview

In the last lecture we talked about finding similarities between texts using TF.IDF, Hamming distance, edit distance and Jaccard Similarity. We also talked about collaborative filtering.

In this lecture we covered: using shingles to create signature for texts and hash functions.

2 Clarification

Hamming distance is a metric telling the amount of different characters at corresponding positions. It is defined only for strings of the same length. This however is rather rare in the real world. The most common workaround is to assume each word is followed by infinite series of meaningless characters. Distance between "hou" and "house" is 2 (as positions 3 and 4 are different)[1].

In mathematical community bags will probably be know as multisets.

3 Collaborative filtering

In last lecture we mentioned three different approaches to collaborative filtering. Those were

- removing films that are not liked
- using two lists (for liked and not liked movies)
- using bags instead sets

We can also think in a little bit different way. Instead bonding users with films, we can bond movie with user. In this approach each movie has users and their marks connected. So eg. movie "Rambo" has 3 users connected to it, each with mark given after watching the movie.

4 Shingles

Sets of shingles can be very big. We want to create short representation of a text. Basically we want to create a signature allowing to recognise similarity of sets.

[1] Once again - computer scientists count from 0 ;P

4.1 Hash function

Definition 4.1. *Hash function is any mapping from a given set (possibly infinite) into a data of fixed size*
eg. SHA-3 (and other cryptographic algorithms), `hashCode()` *in Java used for distinguishing objects.*

In cryptography we often want hash functions that are one-way and collision-free. In practice it's almost impossible to create collision free function, as we may be mapping from infinite set into a finite set.

5 Minhashing

Coming back to our problem from before, let's have a set representation. We can present this set representation using characteristic matrix. It is similar to adjacency matrix of a graph.

MinHash algorithm

1. Pick a random permutation of the rows of the characteristic matrix
2. The minhash value of any column is the number of the first row with 1 (in permuted order)

Fact $J_{SIM}(S_i, S_j) = Pr[h(S_1) = h(S_2)]$

Having two sets S_1, S_2 with signatures $[h_1(S_1), ..., h_n(S_1)]$ and $[h_1(S_2), ..., h_n(S_2)]$, the similarity is computed as

$$\hat{J}_{SIM} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(h_i(S_1) = h_i(S_2)) \tag{1}$$

Expected value can be calculated as

$$E(\hat{J}_{SIM}(S_1, S_2)) = \frac{1}{n} \sum_{i=1}^{n} E(\mathbb{1}(h_i(S_1) = h_i(S_2))) = J_{SIM}(S_1, S_2) \tag{2}$$

Open question Is it possible to have MinHash without random permutation? (Probably not, we're not sure though)