Detection of/between similarity of documents with hashing

Roger Vilaseca Darné, Xavier Lacasa Curto and Xavier Martín Ballesteros
Algorithms

1st December 2018

1 Introduction

2 Jaccard Index

The Jaccard Index, also known as Intersection Over Union (IOU), calculates the percentage of similarity between two sets.

For any pair of sets S and T, the Jaccard Index is defined as:

$$J(S,T) = \frac{|S \cap T|}{|S \cup T|} \tag{1}$$

We can easily deduce that the more common words, the bigger the Jaccard Index, which means that it is more probable that one set is a duplicate of the other.

Example 2.1 *In Figure 2.1 we see two sets S and T. There are 3 elements in their intersection ("I", "love", "chocolate") and 6 in their union ("I", "love", "chocolate", "and", "pizza", "white"). Thus, J(S, T) = 3/6.*

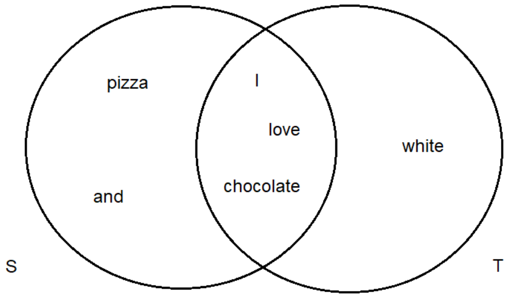

Figure 2.1: Two sets with Jaccard Index 3/6.

3 Shingling of Documents

Any pair of documents can be compared by watching the number of repeated strings they have. The more common strings, the more probable is that one is a duplicate from the other. One way to represent a document as a set is to insert in the set each string that appears in it. If we do so, then duplicated documents that have reorganized the sentences or even the entire text will have plenty of common strings, and will be caught.

3.1 k-Shingles

The idea is not to insert in the set all the words, but a set of characters of size k. Thus, each element of the set will have the same size as the others.

The question now is how big k should be? If we take a small value of k, this will result in many shingles that are present in all documents. Suppose we choose the extreme case ($k = 1$). Then, all documents would result to be similar, as the most used characters are present in all documents. However, if we take a big value of k, then any pair of documents would not share a shingle.

The value of k depends on the size of the documents. A poem will not have the same k value than an article. Otherwise, we could have the problems mentioned before. The important idea in order to choose a good k is: *k should be picked large enough that the probability of any given shingle appearing in any given document is low.*

4 Intro to Minhashing (nom?)

If we succeed in shingling the documents using k shingles, we only will have to compare all pairs of documents using the Jaccard Index and say if there is similarity between them or not. By doing this, we have two problems: time and space complexity.

4.1 Time Complexity

Imagine we have n documents. Then, we have to compare each document with all the rest. Thus, the number of comparisons we have to do is $n * (n-1)/2$ which is equal to $O(n^2)$.

Example 4.1 *Suppose we have 1 million documents. The number of comparisons would be* $5 * 10^{11}$ *which is a huge number.*

$$\frac{(1 * 10^6) * 999.999}{2} = 499.999,5 * 10^6 \approx 5 * 10^{11} \tag{2}$$

4.2 Space Complexity

In typical applications the matrix is sparse, which means that there are more 0s than 1s. We can demonstrate this by calculating the probability of an element of the set to belong to a d

If we take k shingles, then the document have relatively few of the possible shingles. Another way to think about this is with the toys in Christmas Day. Usually, kids would like to have a specific toy, which is very popular at that moment. Then, lots of toys would not be buyed for (a kid/anyone)?.

5 Locality Sensitive Hashing (LSH)

Mas texto.

5.1 Referencies

https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134
https://santhoshhari.github.io/Locality-Sensitive-Hashing/
https://www.youtube.com/watch?v=96WOGPUgMfw
https://www.youtube.com/watch?v=_1D35bN95Go
https://medium.com/engineering-brainly/locality-sensitive-hashing-explained-304e
http://www.mit.edu/~andoni/LSH/
http://infolab.stanford.edu/~ullman/mmds/ch3.pdf

References

[1] Author, *Title*, Editor, (year)