Assignment 3 - Document Similarity and Hashing*

Due: Monday, February 20
Late assignments accepted until Wednesday, February 22
Turn in a hard copy at the start of class

Overview

In this assignment you will explore the use of shingling, Jaccard distance, min hashing, and LSH in the context of document similarity.

You will use four text documents for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/D4.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/D4.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/D4.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/D4.txt`

As usual, it is highly recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory: `http://www.cs.utah.edu/~jeffp/teaching/latex/`

1 Shingling

You will construct several types of k-shingles for all documents. All documents only have at most 27 characters: all lower case letters and space.

[S1] Construct 5-shingles based on characters, for all documents.

[S2] Construct 8-shingles based on characters, for all documents.

[S3] Construct 4-shingles based on words, for all documents.

Remember, that you should only store each shingle once, duplicates are ignored.

A: How many distinct shingles are there for each document with each type of shingle?

B: Compute the Jaccard distance between all pairs of documents for each type of shingling. You should report $3 \times 6 = 18$ different numbers.

2 Min Hashing

We will consider a hash family $\mathcal{H}$ so that any hash function $h \in \mathcal{H}$ maps from $h : \{k\text{-shingles}\} \to [m]$ for $m = 40127$. (was 8191) You are free to choose any hash function you want. If you want to implement your own, let $h_\alpha(x) = \alpha x + 1 \mod m$ where α is a random number in $[m]$, and let x represent the index of each k-shingle.

A: Using shingles S2, build a min-hash signature for each document using $t = \{10, 50, 100, 300, 600\}$ hash functions. For each value of t report the normalized L_0 (had said L_1) distance between all pairs of the 4 documents, estimating the Jaccard distance:

$$\text{normalized-}L_0(a,b) = \frac{1}{t}\sum_{i=1}^{t}\begin{cases}1 & \text{if } a_i = b_i\\ 0 & \text{if } a_i \neq b_i\end{cases}$$

You should report $5 \times 6 = 30$ numbers.

B: What seems to be a good value for t? You may run more experiments. Justify your answer in terms of both accuracy and time.

3 LSH

Consider computing an LSH using $m = 100$ hash functions. We want to find all documents which have Jaccard similarity above $\tau = .25$.

A: Use the trick mentioned in class and the book to estimate the best values of rows r in each of b blocks to provide the S-curve

$$S(s) = 1 - (1 - s^r)^b$$

an early version had the incorrect formula here: $S(s) = (1 - (1 - s)^r)^b$

with good separation at τ.

B: Using your choice of r and b and $S(\cdot)$, what is the probability that you will need to check the exact Jaccard similarity of each pair of the four documents using S2 for having similarity greater that τ?

4 BONUS

Given a set of m independent hash functions $\{h_1, h_2, \ldots, h_m\}$, we can construct a locality sensitive hash function by combining them using AND or OR constructs.

Initially two documents x and y have a probability of collision (indicating they are candidates to compute the exact distance) equivalent to $s = \mathsf{Jac}(x, y)$.

- The r-AND construct provides a new function

  $$f_r(s) = s^r$$

  at the cost of a factor r more hash functions.
- The b-OR construct provides a new function

  $$g_b(s) = 1 - (1 - s)^b$$

  at the cost of a factor b more hast functions.

All notes so far describe LSH constructs of the form

$$g_b(f_r(s)) = 1 - (1 - s^r)^b,$$

at the cost of $r \times b$ more hash functions. However, these functions can be cascaded in any order as many times as needed.

Either prove that a $g_b(f_r(\cdot))$ construct is always optimal for min hashing, or if it is not, show an example of a cascade other than some $g_b \circ f_r$ provides a better separation for a specific set of Jaccard similarities.