similarity search reading

wbg231

January 2023

1 finding nearest neighbors

- this reading is cool, but has some unneeded section

Applications of Near-Neighbor Search

- the Jaccard similarity of sets S and T is given by

$$SIM(S,T) = \frac{|S \cap T||}{|S \cup T|}$$

  that is the ratio of there intersection over there union

- this can be used to find the similarity between two documents on the characters not contextual level
- could be helpful to find plagiarism or identify bad articles from the same source
- another case where Jaccard similarity could be useful is collaborative filtering that is the task of recommending users it's based off of the items liked by other users.

shingling of documents

- k-shingles are the set of strings of length k that appear in a document
- picking the right number of shingles is critical for how useful they are
- instead of working with the shingles directly we hash them, and work at treat the key number they were hashed to as representative of that shingle
- this allows for quick and easy data compression
- could also look at words in a document as shingles

similarity preserving summary of sets

- sets of shingles are really large, even hashing them the space needed to store four byte shingles is the same as that taken to store the document

Element	S_1	S_2	S_3	S_4
a	1	0	0	1
b	0	0	1	0
c	0	1	0	1
d	1	0	1	1
e	0	0	1	0

- we can represent sets of characters using a characteristic or one hot encoding matrix.
- this is not actually how we store the data, but it is a good representation to keep in mind

min hashing

- to minhash a set represented as a column in a characteristic matrix, we first pick a permutation of the rows, the minhash alue of any column is the number of the first row in the permuted order in which the column has 1,
- supose we permuted the table above to be bedac the hash of h(S1)=a h(s2)= c h(S3)=b h(s4)=a
- a cool fact is that the probability that the minhash function for a random permutation of the rows produces the same value for two sets equals the Jaccard similarity of those two sets
- so in other words if we can compute the min hash of a random permutation of the charictersistc matrix fo the sets then we can aproximate the Jaccard similarity of the sets without a lot of storage

min hash singnatures

- so we can represent the charictersistc matrix of a set of vecotrs by icking n random permuations of the the M rows

- then we minhash theose permuations, then for the col represenign S we construct the min hash signuatre for S as the vector of all the miniahses of s
- thus we can think of hhe matrix m as the matrix with all signuatre vectors in its columns
- notice this compreses our repsrensaiton matrix
- we can not do the minhash in pracitce
- but we can simulate the effect of a rnadom perimuate by a random hash function that maps riw numebr to bockets
- so this hash fucntion aproximates a permuation
- so instead of picking n random permuations we pick n random hash functions on teh rows.
- we cosntruct the signuatre matrix by consdering each row in their given order.
- we hande lor r bt computing all hased for row r then for each col if c has a 0 in ror r do ntohing, cif c has a 1 in row r set sig(i,c) to be the min of the current value and the hash of every tow at h
- i am a bit confused about the compuation of the min hashing

local sensitive hashing for documents

- in cases with really large number of docuemtns or itemrs it may be inphesable to even compare the signuatre maytrix
- this is where Local sensetive hasing comes in
- we can just get the most similar items using local senative hashing

LSH for minihash singnatures

- a genearl aproach to lsh is to has items several times in such a way that similar item are more lickly to be hashed to the same bucket than dissimilar items
- then we just check the similarity of pairs that hash to the same bucket

distance metrics

- there are a number of other worth while distance metric
- l2 distance
- jacard distance (1-jackard similarity)
- cosine distance which i proprtional to teh angle between two vectors
- edit distance is how many insertions or deletions must be made to get from one string ot the other
- hamming distance number of components in which two vectors differ
-