Andrew Baumann, Tony Zheng

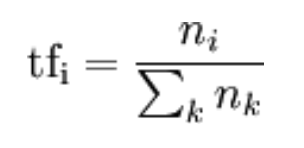
CSCI 493 Big Data Project 3 Report

Professor Lei Xie

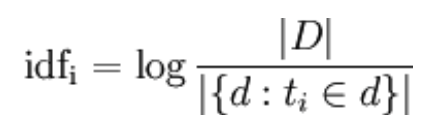
May 15, 2015

For the third project, our assignment was to compute the semantic similarity between all of the terms in the documents. We are given a file where each line represents a document which contains terms encapsulated within a gene tag (gene\_((.)+ + ( ))+\_gene). In order to compute the semantic similarity, there are sub problems that must be solved prior to doing so. These sub problems consist of computing the term frequency, inverse document frequency, and the term frequency–inverse document frequency.

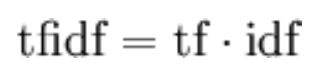
The documents in the file can be separated by the document id and the terms by separating the tab and white space. The result is an RDD of tuples containing the document ids as well as an array type of all the terms pertaining to that document. We first map and compute the term frequency, which is the total number of times a term appears in a document, divided by the number of terms in the documents.



Then to compute the inverse document frequency, we first map all genes as (gene, 1) pairs. We reduce the duplicates in every document, so if a gene X appears in a document twice, it's only given one (gene, 1) pair in the new map. After that, we reduce on 1, pairing all genes and their number of document appearances.



The next step we take the logarithm of (the number of documents / number of times a term appears in the document)



For the final step, we take all of the term frequency-inverse document frequency and compute the Cartesian product, to map all terms together for the semantic similarity. Then we compute the cosine angle of every word. For the numerator of the formula, we use a method to multiply all the term frequency-inverse document frequency of two genes that appear in the same document. Then, we compute the bottom half of the formula using another method. Finally, we filter out all zeroes and then sort on the cosine angle

