CS435: Introduction to Big Data

GTA: Bibek R. Shrestha
Email: cs435@cs.colostate.edu

March 02, 2018
Today...

- Discussion on Programming Assignment 2
- Quizzes Review
Term Frequency (TF) Calculation

- Term Frequency (TF)

\[ TF_{ij} = 0.5 + 0.5 \left( \frac{f_{ij}}{\max_k f_{kj}} \right) \]

- Suppose we have a collection of documents written by \( M \) authors. This collection of documents may contain multiple books written by an author. Let's define a set of documents written by same author as a sub-collection \( j \).

- We define \( f_{ij} \) to be the frequency (Number of occurrences) of term (word) \( i \) in sub-collection \( j \).

- And, \( \max_k f_{kj} \) is the maximum raw frequency of any term \( k \) in the sub-collection \( j \).

- The most frequent term in the sub-collection will have a augmented TF value of 1.
Inverted Document Frequency (IDF) Calculation

- Suppose that term $i$ appears in $n_{ij}$ sub-collections within the corpus. Then,

$$IDF_i = \log_{10}(N/n_i)$$

- where, $N$ is the total number of sub-collections (number of authors).
TF-IDF Calculation

- The TF-IDF score is defined as:

\[ TF_{ij} \times IDF_i \]

- The terms with the highest **TF-IDF** score are considered the best words that characterize the document.
Author Attribute Vector

- The result of performing calculations above is that each set of books written by the same author in your corpus will have an Author Attribute Vector (AAV),

\[ AAV_m = (TF.IDF_{word1}, TF.IDF_{word2}, TF.IDF_{word3}, \ldots, TF.IDF_m) \]
Cosine-Distance Calculation

- The AAVs will be used to calculate the Cosine Distance to measure the similarity between the authors’ writing styles.
- Suppose that we have two authors with vectors,
  \[ AAV_1 = [x_1, x_2, \ldots, x_m] \text{ and } AAV_2 = [y_1, y_2, \ldots, y_m]. \]
- The Cosine Similarity between them is defined as,
  \[
  \text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}
  \]
  where, \( A_i \) and \( B_i \) are components of \( AAV_1 \) and \( AAV_2 \) respectively.
- Finally, this approach will be able to provide a ranked list of authors (top 10 most similar authors) for a document with unknown authorship.
How to Approach PA2?

- Recall what we did in PA1.
- For PA2, first calculate frequency of words per Author. (this is Profile 3 in PA1)
- Use the previous output to calculate TF for each of the word. You will need to know the value of $\max_k f_{kj}$ beforehand. [$TF_{ij} = 0.5 + 0.5 \left( \frac{f_{ij}}{\max_k f_{kj}} \right)$]
- Now, to calculate IDF, you need $n_i$ where $n_i$ is total occurrence of the word. (this is Profile1 in PA1) [$IDF_i = \log_{10}(N/n_i)$]
continue...

- Finally, $TF*IDF$ value for each word per author.
Things to remember

- You will use multiple jobs and will have to swap key value pairs as required. (similar to Profile 2 in PA1, but use of compositeKey not required.)
- You will be able to calculate IDF and TF-IDF in a single job
- You will also have to write separate MapReduce jobs to test dataset (with unknown author) and calculate cosine distance to identify top 10 closest match for the test file.
Quizzes

- Discussion
- Midterm sample questions
In the next recitation...

- Finish discussion on Programming Assignment 2.
- Go through common issues that might occur while writing the program.