Email Classification
Team Ravana

Team Members
Cliffton Fernandes
Nikhil Keswaney
Hello!

Cliffton Fernandes
MS-CS

Nikhil Keswaney
MS-CS
Email Classification

Spam: Unsolicited Junk Mail

Ham: Desired emails and not spam.

Accuracy improves with increase in processed data.

Enron Corpus (500k Emails)

Large amount of computational resources and time.

Parallel computing can be used to reduce the amount of time.
Approach

- K - Nearest Neighbours Algorithm.
- Classification algorithm.

Example of KNN.
How to do it?

● Implementing KNN (K nearest Neighbour) using TF-IDF (Term Frequency - Inverse Document Frequency) for weight matrix calculation.

● Benchmarking the implemented technique.
TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency)

● What is TF-IDF?
  ▪ TDIDF is a numerical statistic method which allows the determination of weight for each term in each document.
  ▪ This method is often used in NLP or in information retrieval and text mining.
  ▪ This method evaluates importance of terms (Words) in document collection.
  ▪ The importance of the text is increased proportionally to the number of times it appears in the documents.
TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency)

● **Term Frequency**
  The method computes the frequency of a particular word $i$ in that particular document $j$.

  \[
  f_{ij} = \text{Frequency of term } i \text{ in document } j.
  \]
  \[
  tf_{ij} = \text{Term Frequency of } i \text{ in document } j.
  \]
  \[
  tf_{ij} = \frac{f_{ij}}{\max \{f_{ij}\}}
  \]
TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency)

- **Inverse Document Frequency**
  
  IDF method determines the relative frequency of words in a specific document through an inverse proportion of the word over the entire document corpus.

\[
idf_i = \text{Inverse document frequency.} \\
df_i = \text{Document Frequency of term i.} \\
N = \text{number of documents.}
\]

\[
idf_i = \log_2 \frac{N}{df_i}
\]
TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency)

\[ f_{ij} = \text{Frequency of term } i \text{ in document } j. \]
\[ tf_{ij} = \text{Term Frequency of } i \text{ in document } j. \]
\[ idf_i = \text{Inverse document frequency.} \]
\[ df_{ij} = \text{Document Frequency of term } i \text{ in all documents.} \]
\[ N = \text{number of documents.} \]

\[ (TF - IDF)_{ij} = \frac{f_{ij}}{\max\{f_{ij}\}} \times \log_2 \frac{N}{df_i} \]
example of a TF-IDF matrix.

1 – Spam
0 – Ham
2 – Unclassified

<table>
<thead>
<tr>
<th>Email</th>
<th>Cheap</th>
<th>Purchase</th>
<th>Contract</th>
<th>...</th>
<th>Submitted</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.23</td>
<td>0.12</td>
<td>0</td>
<td>...</td>
<td>9.45</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.22</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>7.1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
<td>...</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0.0012</td>
<td>11.22</td>
<td>0</td>
<td></td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>500000</td>
<td>7.22</td>
<td>4.11</td>
<td>0</td>
<td></td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
example of a TF-IDF matrix.

Similarity of two emails is how close i.e. how many words it has in common.

To classify email 4 we find the similarity between that email and all the other classified emails.

Example:

\[
\begin{align*}
similarity(4,1) &= 2.11 \\
similarity(4,2) &= 20.00 \\
similarity(4,3) &= 13.00
\end{align*}
\]

So from the above similarity scores we can see that email 4 is closest to 1 and 3 so we mark it as **SPAM** and change its answer to 1.
Sequential program
START

LOAD DATA
(OF THE CLASSIFIED AS WELL AS UNCLASSIFIED)

MAKE TF-IDF SCORES
(OF THE CLASSIFIED AS WELL AS UNCLASSIFIED)

SELECT WORDS WITH HIGHEST TFIDF SCORES
(BUT THIS WILL BE TAKEN FROM ONLY CLASSIFIED)

START CLASSIFICATION OF UNCLASSIFIED EMAIL

FIND THE DISTANCE OF THE EMAIL TO CLASSIFY TO ALL
THE CLASSIFIED EMAIL.

FIND THE K-Nearest Neighbours AND CLASSIFY THE EMAIL
TO THE IDENTITY OF ITS MAXIMUM NEIGHBOURS.

ANY MORE EMAILS LEFT TO CLASSIFY?

YES

WRITE BACK THE UNCLASSIFIED EMAILS TO A FILE WITH
THEIR PREDICTED CLASS

NO

END
Parallel program
START

LOAD DATA
(OF THE CLASSIFIED AS WELL AS UNCLASSIFIED DATASET)

MAKE TF-IDF SCORES
(OF THE CLASSIFIED AS WELL AS UNCLASSIFIED DATASET)

SELECT WORDS WITH HIGHEST TFIDF SCORES
(BUT THIS WILL BE TAKEN FROM ONLY CLASSIFIED)

START CLASSIFICATION OF THE UNCLASSIFIED EMAIL

FIND THE DISTANCE OF THE EMAIL TO CLASSIFY TO ALL THE CLASSIFIED EMAIL

FIND THE K-Nearest Neighbours AND CLASSIFY THE EMAIL TO THE IDENTITY OF ITS MAXIMUM NEIGHBOURS.

FIND THE DISTANCE OF THE EMAIL TO CLASSIFY TO ALL THE CLASSIFIED EMAIL

FIND THE K-Nearest Neighbours AND CLASSIFY THE EMAIL TO THE IDENTITY OF ITS MAXIMUM NEIGHBOURS.

FIND THE DISTANCE OF THE EMAIL TO CLASSIFY TO ALL THE CLASSIFIED EMAIL

FIND THE K-Nearest Neighbours AND CLASSIFY THE EMAIL TO THE IDENTITY OF ITS MAXIMUM NEIGHBOURS.

ANY MORE EMAILS LEFT TO CLASSIFY?

YES

WRITE BACK THE UNCLASSIFIED EMAILS TO A FILE WITH THEIR PREDICTED CLASS

END

NO
DEMONSTRATION.
Thanks!

Any questions?