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.
Literature Search Paper 1

- **Title:** A Novel Method for Detecting Spam Email using KNN Classification with Spearman Correlation as Distance Measure.

- **Authors:** Ajay Sharma, Anil Suryawanshi.

- **International Journal of Computer Applications**

- **Year of Publication:** 2016

- **Link:** https://www.ijcaonline.org/archives/volume136/number6/24159-2016908471
Problem Addressed

- K-NN finds a similarity measure between the objects for classification.
- Finding the right similarity measure effects the accuracy and speed of the classification.
- Comparative study of Euclidean distance vs Spearman's Coefficient.
Novel Contributions
Similarity Measures

Euclidean Distance

\[ \text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \]

\[ X_1 = (x_{11}, x_{12}, \ldots, x_{1n}) \]

\[ X_2 = (x_{21}, x_{22}, \ldots, x_{2n}) \]

Spearman’s Coefficient

\[ d_{ij} = 1 - \frac{6 \sum_{i=1}^{n} (\text{rank}(x_i) - \text{rank}(y_j))^2}{n(n^2 - 1)} \]

n = Number of objects.
Evaluation Metrics

- F-Measure
- Precision
- Recall
- Accuracy

- TP : True Positive
- TN : True Negative
- FP : False Positive
- FN : False Negative

\[ F - measure = \frac{2 \times P \times R}{P + R} \]

Where \( P \) and \( R \) are defined as:

\[ P(\text{precision}) = \frac{TP}{TP + FP} \]

\[ R(\text{recall}) = \frac{TP}{TP + FN} \]

\[ \text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \]
<table>
<thead>
<tr>
<th>Classification Technique</th>
<th>Train-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50-50</td>
</tr>
<tr>
<td>KNN with Euclidean</td>
<td>0.9290</td>
</tr>
<tr>
<td>KNN with Spearman</td>
<td>0.9772</td>
</tr>
</tbody>
</table>

**Figure 7: KNN_e vs KNN_s (Precision)**
<table>
<thead>
<tr>
<th>Classification Technique</th>
<th>Train-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50-50</td>
</tr>
<tr>
<td>KNN with Euclidean</td>
<td>0.3771</td>
</tr>
<tr>
<td>KNN with Spearman</td>
<td>0.9352</td>
</tr>
</tbody>
</table>

**Figure 8: KNN e vs KNNs (Recall)**
<table>
<thead>
<tr>
<th>Classification Technique</th>
<th>Train-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50-50</td>
</tr>
<tr>
<td>KNN with Euclidean</td>
<td>0.5364</td>
</tr>
<tr>
<td>KNN with Spearman</td>
<td>0.9560</td>
</tr>
</tbody>
</table>

Figure 9: KNNe vs KNNs (F-Measure)
<table>
<thead>
<tr>
<th>Classification Technique</th>
<th>Train-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50-50</td>
</tr>
<tr>
<td>KNN with Euclidean</td>
<td>0.4491</td>
</tr>
<tr>
<td>KNN with Spearman</td>
<td>0.9691</td>
</tr>
</tbody>
</table>

Figure 10: KNNe vs KNNs (Accuracy)
Applications to our Investigation

- The paper concludes that the Spearman’s coefficient is a better performing similarity coefficient for the K-NN algorithm.

- Therefore we will use the Spearman’s coefficient in our project.
Title: KNN with TF-IDF Based Framework for Text Categorization.

Authors: Bruno Trstenjak, Sasa Mikac, Dzenana Donko.

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Year of Publication: 2013

Link: https://doi.org/10.1016/j.proeng.2014.03.129
Problems Addressed

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

- Benchmarking the implemented technique.
Novel Contributions

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 appearing in the documents.
Novel Contributions

TF-IDF (Term Frequency - Inverse Document Frequency)

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

\[
\begin{align*}
  f_{ij} &= \text{Frequency of term } i \text{ in document } j. \\
  tf_{ij} &= \text{Term Frequency of } i \text{ in document } j. \\
  tf_{ij} &= \frac{f_{ij}}{\max \{f_{ij}\}}
\end{align*}
\]
Novel Contributions

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.

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

\[
idf_i = \log_2 \frac{N}{df_i}
\]
Novel Contributions

TF-IDF (Term Frequency - Inverse Document Frequency)

\[ f_{ij} = \text{Frequency of term } i \text{ in document } j. \]
\[ \text{tf}_{ij} = \text{Term Frequency of } i \text{ in document } j. \]
\[ \text{idf}_{ij} = \text{Inverse document frequency}. \]
\[ \text{df}_{ij} = \text{Document Frequency of term } i \text{ in document } j. \]
\[ N = \text{number of documents}. \]

\[ (TF - IDF)_{ij} = \frac{f_{ij}}{\max\{f_{ij}\}} \times \log_2 \frac{N}{\text{df}_i} \]
Novel Contributions

TF-IDF (Term Frequency - Inverse Document Frequency) Benchmarks
Novel Contributions

TF-IDF (Term Frequency - Inverse Document Frequency) Benchmarks

- The last test was focused on measuring the accuracy of classification depending on the document category.
- The authors suspected that it was because of the preprocessing of the document i.e. removing undesired characters and words.
Applications to our Investigation

- Term Binary (As explained in our previous presentation)

Spam = \{ \text{cheap} : 40, \text{purchase} : 33, \ldots \}
Ham = \{ \text{contract} : 20, \text{submitted} : 11, \ldots \}

<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>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>Spam</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td>Ham</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
<td>Ham</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>Spam</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>500000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>Spam</td>
</tr>
</tbody>
</table>
Literature Search Paper 3

- **Title:** Improved KNN Text Classification Algorithm with MapReduce Implementation.
- **Authors:** Yan Zhao, Yun Qian, Cuixia Li.
- **The 2017 4th International Conference on Systems and Informatics (ICSAI 2017)**
- **Year of Publication:** 2017
- **Link:** https://ieeexplore.ieee.org/document/8248509
Problems Addressed

● Implementing an improved TF-IDF (Term Frequency - Inverse Document Frequency) for weight matrix calculation.

● Implement the KNN algorithm with the improved TF-IDF using Map-Reduce.

● Benchmarking the implemented technique
Novel Contributions

● Improvements
  ▪ In order to reduce the matrix size they only selected P words with the highest TF-IDF scores and removing the duplicate words.
  ▪ In M documents the matrix created would be of size $M \times P$.
  ▪ The words with the highest scores are called Feature vectors.
Novel Contributions

Traditional TF-IDF considering $i$ as the word in the $j$ document.

$$(TF - IDF)_{ij} = \frac{f_{ij}}{\max \{f_{ij}\}} \times \log_2 \frac{N}{df_i}$$

● Improvements

- When we calculate the IDF factor of a word it reduces the value of the word if it occurs a lot in all the documents.
- What if those words occur in more concentration in one category?

They included a factor

$$DF_i = \text{Discrimination Factor of a word } i$$

to increase the weight of the word that should get increased

$$T = \text{Total number of words in this category.}$$

$$m_i = \text{number of the text in the category.}$$

$$DF_i = \log_2 \frac{T}{m_i}$$

$$(TF - IDF - DF)_{ij} = (TF - IDF)_{ij} \times DF_i$$
Novel Contributions

● Improvements
  ▪ Some words are distributed more evenly across the dataset. For example, if China appears 9 times in spam and 10 times in ham, then that should be eliminated while making feature vectors.
  ▪ But because of TF-IDF, it will have a high value and would be included.
  ▪ So they used the below standard deviation formulae to remove all the evenly distributed words. From the feature vector.

\[ G = \text{Total number of categories.} \]
\[ x_j = \text{number of times that word occurred in that category.} \]
\[ \mu = \text{Sum of times that word occurred in all categories.} \]

\[
\sigma = \sqrt{\frac{1}{G} \sum_{j=1}^{G} \left( \frac{x_j - \mu}{G \times \mu} \right)^2}
\]
Novel Contributions

- **Implementation in MAP-REDUCE.**
  - KNN is a widely used text classification data but when faced with enormous data the sequential program seems to be inadequate.
  - So they used MAP-Reduce.
  The basic idea of the MAP-Reduce version is as follows

1. Calculate the (TF-IDF-DF) score matrix According to the formulae.
2. Obtain Feature Vectors.
3. Vectorize training sets.

**MAP:**
  Calculate the distance of that vector with all the training data vectors.

**REDUCE:**
  Merge the results of the MAP outputs find the K nearest neighbours and assign the category that it has maximum neighbours with.
Novel Contributions

Benchmarks
Applications to our Investigation

- Since the Categories we are making are only 2 (Spam and Ham) so the standard deviation formulae becomes
  \[ \sigma = \sqrt{\frac{1}{2} \sum_{j=1}^{2} \left( \frac{x_j - \mu}{G \times \mu} \right)^2} \]

- The Weight Matrix that we are calculation will include the modifications.
- The implementation of the Parallel algorithm will be using the MAP-REDUCE functionality with modifications.
  Modifications: The distance calculation of the emails will be done using the Spearman's Coefficient.
Thanks!

Any questions?