Detecting Fake News with Supervised Learning

Benjamin R. Perucco

December 30, 2020

1 Definition

1.1 Project Overview

1.2 Problem Statement

1.3 Metrics

2 Analysis

2.1 Algorithms and Techniques

News articles must be converted into a structured, mathematical representation in order be applied by machine learning techniques. The following chapters disucss how we can convert text into numerical features.

2.1.1 n-gram Model

The n-gram is usually defined a contiguous sequence of words with length n. For example, if $n = 1$, we speak of a unigram that contains only single word tokens. Or if $n = 2$, we denote this as a bigram which is built on two adjacent word tokens.

Consider the following text: "Sometimes we eat green apples, and sometimes, the apples we eat are red." Based on a unigram (1-gram), we obtain a set of tokens: {'sometimes', 'we', 'eat', 'apples', 'green', 'and', 'the', 'are', 'red'}. We can derive a frequency array of tokens in the text: [2, 2, 2, 2, 1, 1, 1, 1, 1]. For the bigram (2-gram), another set of tokens is obtained: {'sometimes we', 'we eat', 'eat green', 'green apples', 'apples and', 'and sometimes', 'sometimes the', 'the apples', 'apples we', 'eat are', 'are red'}. The corresponding frequency array of tokens in the text is: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]. Please note that punctuation is not considered when converting text into tokens.

In order to build frequency arrays for a set of texts (or documents), a common vocabulary needs to be built of which the n-gram model is underlying principle.

2.1.2 Vocabulary

Consider a corpus D which contains a set of documents $\{d_1, d_2, d_3, ..., d_n\}$. Then a vocubulary F is a set of tokens $\{f_1, f_2, f_3, ..., f_m\}$ extracted from the corpus D. Please remind yourself that a token is created based on the n-gram model. Usually for a set of tokens, only the m mostly occuring tokens in a corpus D are considered. In the following, the tokens are denoted as features as these build the features (or independent variables) of a machine learning model.

2.1.3 Definitions

Let $\sigma(f_j, d_i)$ denote the number of occurances of feature f_j in document d_i. Then a feature matrix X can be built, where

$$X = \begin{bmatrix} \sigma(f_1, d_1) & \sigma(f_2, d_1) & \sigma(f_3, d_1) & ... & \sigma(f_m, d_1) \\ \sigma(f_1, d_2) & \sigma(f_2, d_2) & \sigma(f_3, d_2) & ... & \sigma(f_m, d_2) \\ \sigma(f_1, d_3) & \sigma(f_2, d_3) & \sigma(f_3, d_3) & ... & \sigma(f_m, d_3) \\ ... & ... & ... & ... & ... \\ \sigma(f_1, d_n) & \sigma(f_2, d_n) & \sigma(f_3, d_n) & ... & \sigma(f_m, d_n) \end{bmatrix}. \tag{1}$$

An element $\sigma(f_j, d_i)$ in X (representing the document d_i and feature f_j) is abbreviated using the notation σ_{ij} for simplicity.

2.1.4 Term Frequency Model

The matrix X could be already used for machine learning. Features usually need to be normalized in machine learning to increase performance. Therefore, the term frequency (TF) model normalizes matrix X to $\hat{X}$. An element $\hat{\sigma}_{ij}$ of $\hat{X}$ is written as

$$\hat{\sigma}_{ij} = \frac{\sigma_{ij}}{\sum_{j=1}^{m} \sigma_{ij}}. \tag{2}$$

Or spoken in plain language: the number of occurances of a token f_j in a document d_i is divided by the total number of occurances of all tokens $\{f_1, f_2, f_3, ..., f_m\}$ in the same document d_i. So we end up with a representation where the importance of each feature can be compared to other features in the same document.

2.1.5 Inverse Document Frequency Model

The inverse document frequency (IDF) model is used to define the feature importance not just in a document d_i but also compare its importance within a corpus D. A matrix Y is introduced, where an element δ_{ij} of Y is 1 if $\hat{\sigma}_{ij} > 0$. An inverse normalization is performed on the matrix Y resulting in a vector $\hat{y}$. An element $\hat{\delta}_j$ of $\hat{y}$ is

$$\hat{\delta}_j = 1 + \log\left[\frac{|D|}{\sum_{i=1}^{n} \delta_{ij}}\right]. \tag{3}$$

Or spoken in plain language: in a corpus D which comprises of a set of documents $\{d_1, d_2, d_3, ..., d_n\}$, its is counted in how many documents the feature f_j appears. This number is used to divide the number of documents $|D|$ in a corpus D. Consider two examples: if a feature f_j occurs in each document $\{d_1, d_2, d_3, ..., d_n\}$, we divide the number of documents $|D|$ in a corpus D by the same number. So expression 3 results in 1, weighting feature f_j as 1. On the other hand, if a feature f_j occurs only in one document, expression 3 produces a much larger number, thus increasing the weight of feature f_j in the corpus D.

Finally, a matrix Z is obtained as the element-wise product of the term frequency matrix $\hat{X}$ and the inverse document frequency vector $\hat{y}$ ($Z = \hat{X} \odot \hat{y}$). This is written for an element z_{ij} as

$$z_{ij} = \hat{\sigma}_{ij} \cdot \hat{\delta}_j, \quad \text{element-wise for j = 1, 2, 3, ..., m} \tag{4}$$

This is denoted as term frequency inverse document frequency model (TF-IDF).