ISA 414 – Managing Big Data

Lecture 14 – Text Mining (Part I)

Dr. Arthur Carvalho
arthur.carvalho@miamioh.edu

MIAMI UNIVERSITY

Lecture Objectives

- Learn how to prepare textual data for analysis
  - Bag of words approach
    - Key concepts
      - Term Frequency (TF)
      - Inverse Document Frequency (IDF)
      - Term Frequency Inverse Document Frequency (TFIDF)
    - Text cleaning
      - Stemming
      - Stop words

Lecture Instructions

- Download the notebook "*Lecture 14.ipynb*" available on Canvas
  - Open the above file with VS Code

Data Analysis

- Previous lectures
  - Data analytics problem: classification and regression
  - Modeling: decision trees, random forests
- Assumption in our analysis
  - Data are in tabular format
- What if the data are unstructured (*e.g.*, textual)?
  - Can we still build models to make predictions?

Data Analysis

- Text is different
  - No predefined, uniformly applicable format
    - *E.g.*, contrast a tweet against an article on the NY Times
  - Context dependent
    - *E.g.*, 'ring' can be a verb or a noun; 'incredible' can have positive or negative qualification; kkk = laughing in Portuguese
  - Spelling mistakes and grammatical errors may contaminate texts
    - Homonyms and synonyms increase the complexity of an analysis

Data Analysis

- Text mining (analytics): deriving information from textual data
  - Many different tasks
    - Text categorization (*e.g.*, spam filter – next lecture)
    - Sentiment analysis (next lecture)
    - Text clustering (*e.g.*, plagiarism checker)
    - Automatic summarization (*e.g.*, “consensus” of several reviews)
    - Topic modeling (future classes)
  - Involves extensive preprocessing
    - Oftentimes, a tabular structure is imposed
      - Topic of this lecture
  - Let’s build intuition first

Human Classifier: SPAM or NOT SPAM?

Greetings to you my friend,

I know this will come to you as a surprise because you do not know me.
I am John Alison and I work at Central Bank of Nigeria, packaging and courier department.

I got your contact among others from a search on the internet and I was inspired to seek your co-operation. I want you to help me clear this consignment that is already in Europe which I shipped through our CBN accredited courier agent. The content of the package is $20,000,000 all in $100 bills, but the courier company does not know that the consignment contains money.

All I want you to do for me now is to give me your mailing address, your private phone number, and credit card information so that I can deposit some money to cover your upfront costs.

Please, let me know your response as soon as possible. WE CANNOT WASTE THIS OPPORTUNITY.

With Love,
john_alison444@yahoo.com

Human Classifier: SPAM or NOT SPAM?

Hi Professor Carvalho,

I first wanted to thank you for all the kind feedback we received on our project. We spent a lot of time on it, as I'm sure you did grading them, and I really appreciate your comments.

My grade is currently lingering around a B+ due to the difficulties I had in the first exam. I was wondering if the 90% cut off for an A- is a hard cut off, or if I receive say an 89.9, would that be an A-? I understand every professor has their own grading policies, but I just wanted to know for sure what I needed to score on the final to get an A or an A-.

Thanks again for a great semester, I really appreciate all the time you spend grading our assignments and working with us individually.

Text Mining

- How do you know that the first email is SPAM, but the second is not?
  1. Context
     - Requires deep understanding of the language
  2. Presence of certain keywords
     - $20,000,000, credit, card, love
     - Other common keywords
       - Viagra, sex, chat, money, currency, bitcoin, …
  - Bottom line: simply checking for the presence/absence of certain words can help with the classification task

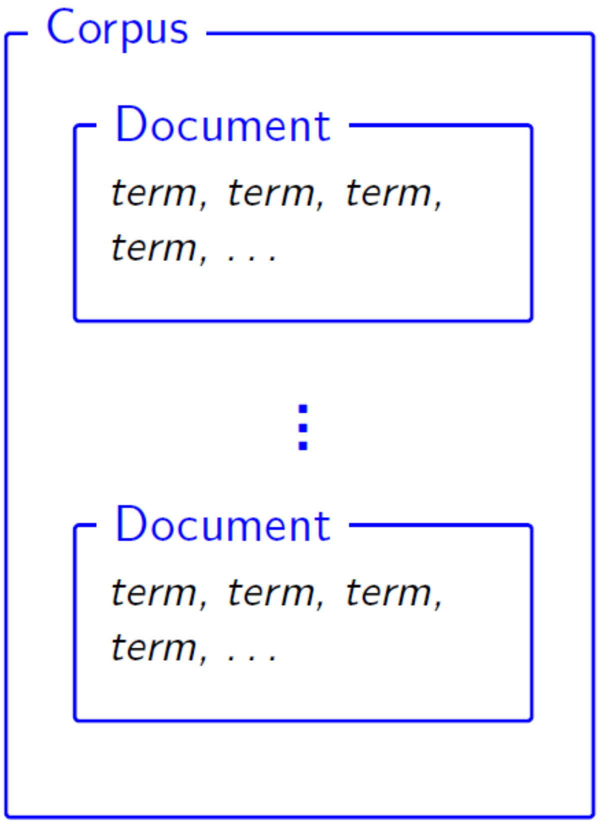
Text Mining

- Based on the previous observation, we have one way of imposing a tabular structure
  - Rows = observations (textual documents)
  - Columns = individual words
  - Cells = some sort frequency measure (counting)
  - Example:

Doc	the	hotel	has	one	bad	room	of	bathroom	is	other	good
1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	0	0	1	1	1	0	1	0	0
3	1	0	0	1	1	0	0	1	1	1	1

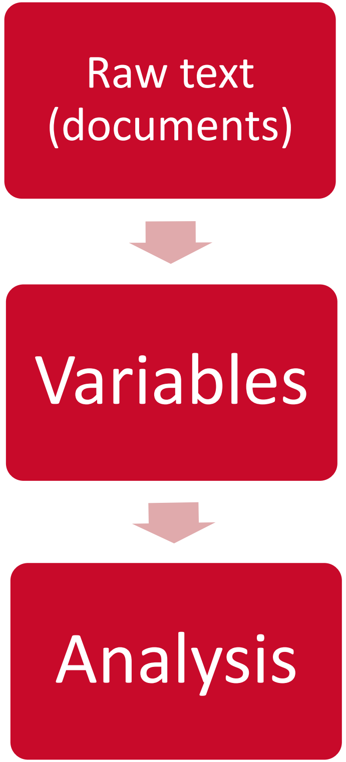
Text Mining

- Terminology taken from the Information Retrieval (IR) domain
  - A piece of text is referred to as a **document**
  - Documents consist of items called **terms**, **tokens** or, plainly, **words**
  - Documents that belong together form a **corpus**
    - In document-oriented databases, like MongoDB, this is called a **collection**



Text Mining

- Our approach in this course
  - Raw documents form the basis of the analysis
  - Preprocessing is done to transform the raw texts into a tabular format
  - The constructed data sets are used in different analyses (next lecture)
    - Descriptive
      - Sentiment analysis
      - Word association
    - Statistical modeling
      - Classification/regression problems



Text Mining

- There are different ways of representing a document
- A well-known representation format is the "**bag of words**"
  - Each term, token, or word in a document is considered individually
    - Only counts of individual words matter
  - The syntax of the language is ignored
    - That is, grammar, word order, sentence structure
  - Simplistic, but still powerful approach

Bag of Words

- Counts of individual words will lead to the familiar tabular format
  - **Document-Term Matrix (DTM)**
  - Different ways of counting words
    - Within documents: **term frequency** (TF)
    - Across documents: **inverse document frequency** (IDF)
    - Within and across documents: **term frequency + inverse document frequency** (TFIDF)

Term Frequency (TF)

Term Frequency

- Term frequency (TF) refers to the occurrence of words in a document
  - Binary TF
  - Frequency-based TF
    - Absolute
    - Normalized

Term Frequency

- Binary term frequencies: indicates whether a term is present
  - All the occurrences of a word in a document count as one
  - Ideal for short documents like tweets
  - Example
    - Doc 1: the hotel has one bad room
    - Doc 2: the room of the hotel is bad
    - Doc 3: one bathroom is bad, the other bathroom is good

Doc	the	hotel	has	one	bad	room	of	bathroom	is	other	good
1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	0	0	1	1	1	0	1	0	0
3	1	0	0	1	1	0	0	1	1	1	1

Term Frequency

- Frequency-based term frequencies
  - Ideal for long documents like product reviews
  - Can be either absolute or normalized
    - **Absolute**: counts the number of occurrences of a word in a document
    - **Normalized**: counts the number of occurrences of a word in a document divided by the number of words in the document
      - Deals with documents of varying length

Term Frequency

- Absolute term frequency
  - Example
    - Doc 1: jazz music has a swing rhythm
    - Doc 2: swing is hard to explain
    - Doc 3: swing rhythm is a natural rhythm

Doc	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
1	1	0	0	1	0	1	1	0	1	1	0
2	0	1	1	0	1	0	0	0	0	1	1
3	1	0	0	0	1	0	0	1	2	1	0

Term Frequency

- Normalized text frequency
  - Example
    - Doc 1 (length 6): jazz music has a swing rhythm
    - Doc 2 (length 5): swing is hard to explain
    - Doc 3 (length 6): swing rhythm is a natural rhythm

Doc	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
1	0.166	0	0	0.166	0	0.166	0.166	0	0.166	0.166	0
2	0	0.2	0.2	0	0.2	0	0	0	0	0.2	0.2
3	0.166	0	0	0	0.166	0	0	0.166	0.332	0.166	0

Inverse Document Frequency (IDF)

Inverse Document Frequency

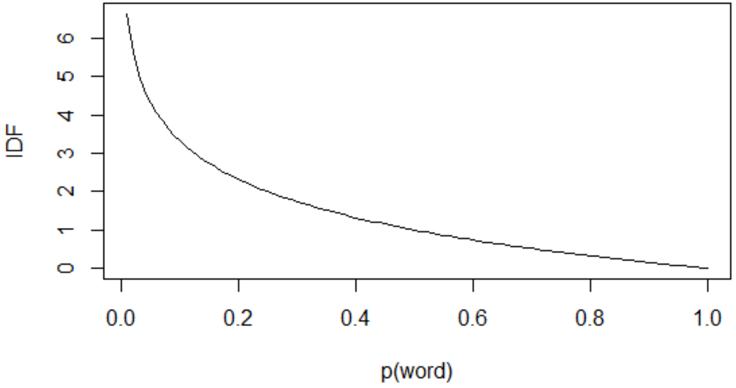
- Term frequencies measure how popular words are inside documents
  - But not across documents
  - Incomplete metric
    - Example: the word “the” is likely to occur many times in many documents
      - Can you classify an email as SPAM based on the presence/absence of the word “the”?
    - Example: the word “Viagra” is likely to occur a few times in only a few documents
      - Is it an important term when classifying emails?

Inverse Document Frequency

- The inverse document frequency (IDF) measures how rare words are in a corpus
  - A term is likely to represent a single document well if the term is rare in the corpus
- Let $p(word) = \frac{\text{Number of documents containing the term "word"}}{\text{total number of documents}}$
  - Fraction of documents that contains the term '*word*'
- $IDF(\text{word}) = \log_2 \frac{1}{p(word)}$

Inverse Document Frequency

- One can interpret IDF as a boost a term gets for being rare
  - Plot
    - Left region: a word receives high IDF score when it is very rare
    - Right region: IDF of very common words (less discriminatory)



Inverse Document Frequency

- Example
  - Doc 1: jazz music has a swing rhythm
  - Doc 2: swing is hard to explain
  - Doc 3: swing rhythm is a natural rhythm

Doc	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
# of docs	2	1	1	1	2	1	1	1	2	3	1
P(t)	0.66	0.33	0.33	0.33	0.66	0.33	0.33	0.33	0.66	1	0.33
IDF	0.58	1.58	1.58	1.58	0.58	1.58	1.58	1.58	0.58	0	1.58

Term Frequency Inverse Document Frequency (TFIDF)

TFIDF

- TF measures the frequency of word occurrences in a document without reference to the corpus

- IDF indicates how rare words are in the corpus

- What about combining these two metrics?
  - TFIDF

TFIDF

- TFIDF(word; document) = TF(word, doc)*IDF(word)
  - IDF reduces the weight of common terms and inflates the weight of rare terms
  - Example: Absolute TFIDF
    - Doc 1: jazz music has a swing rhythm
    - Doc 2: swing is hard to explain
    - Doc 3: swing rhythm is a natural rhythm

	Doc	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
TF	1	1	0	0	1	0	1	1	0	1	1	0
	2	0	1	1	0	1	0	0	0	0	1	1
	3	1	0	0	0	1	0	0	1	2	1	0
IDF		0.58	1.58	1.58	1.58	0.58	1.58	1.58	1.58	0.58	0	1.58

TFIDF

- TFIDF(word; document) = TF(word, doc)*IDF(word)
  - IDF reduces the weight of common terms and inflates the weight of rare terms
  - Example: Absolute TFIDF
    - Doc 1: jazz music has a swing rhythm
    - Doc 2: swing is hard to explain
    - Doc 3: swing rhythm is a natural rhythm

	Doc	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
TFIDF	1	0.58	0	0	1.58	0	1.58	1.58	0	0.58	0	0
	2	0	1.58	1.58	0	0.58	0	0	0	0	0	1.58
	3	0.58	0	0	0	0.58	0	0	1.58	1.16	0	0

TFIDF

- Which counting (frequency metric) should one use?
  - Hard to say without a context
    - TFIDF is one of the most popular metrics
      - Short documents: Binary TF
      - Documents with similar lengths: Absolute TF
      - Documents with drastically different lengths: Normalized TF
  - One can always try all metrics when making predictions!
    - Build one predictive model for each different DTM
    - Evaluate each model
    - Pick the most accurate one

TFIDF in Python

TFIDF

- We shall use the (sub)module feature_extraction inside sklearn

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = [
    'jazz music has a swing rhythm',
    'swing is hard to explain',
    'swing rhythm is a natural rhythm',
]
vectorizer   = TfidfVectorizer()
tfidf_result = vectorizer.fit_transform(corpus)
```

TFIDF

- Important technical points
  - The IDF formula used by sklearn is not the "textbook version"
    - They use smoothing techniques to avoid division by 0
  - The resulting TFIDF values are normalized
    - Technical details available at https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
  - The data structure that stores the TFIDF values is called "sparse matrix"
    - It is a concise way of storing data since most TFIDF values are equal to 0 in practice
    - Look at today's notebook for a sample code on how to convert a sparse matrix to a data frame

Text Mining

- Note that not every single word in your corpus will be part of the DTM
  - By default, sklearn removes words of length one and punctuation marks
    - *E.g.*, “a”, “[!@#$%^&*()]{}:;<>,.?/~`”
  - Moreover, all words are automatically converted to lowercase
  - One can further remove words that are not informative (too common)
    - These are called **stop words**
    - sklearn already comes with a stop-word list; use the argument stop_words = "english“ in the function TfidfVectorizer (see today’s notebook)
- The above steps help decreasing the number of variables
  - Otherwise, a simple corpus can result in thousands of variables

Text Mining

- It is also commonplace to perform **stemming**
  - Stemming: reduce words to their roots
    - *E.g.*, playing -> play
  - Not covered in this course (search for CountVectorizer)
- The 'bag of words' disregards word combinations, like 'bed and breakfast' or 'New York'
  - "BUS" = vehicle; "420" = code for drugs; "BUS 420" = FSB course
  - **N-grams** consider combinations of *n* words
    - Uni-grams, bi-grams, tri-grams, …
    - Advantage: more information is brought to the analysis
    - Disadvantage: number of variables may become very large
    - Look at today's notebook for a sample code

Text Mining

- How can we use document-term matrices for predictive analytics?
  - From unstructured to structured data
    - Doc 1: you won $1000000 dollars (target: SPAM)
    - Doc 2: I love ISA 414 (target: NOT SPAM)
    - Doc 3: improve your life now: buy Viagra (target: SPAM)

Doc	1000000	414	buy	dollars	i	improve	isa	life	love	now	viagra	won	you	your	Target
1	0.4	0	0	0.4	0	0	0	0	0	0	0	0.4	0.4	0	SPAM
2	0	0.4	0	0	0.4	0	0.4	0	0.4	0	0	0	0	0	NOT SPAM
3	0	0	0.26	0	0	0.26	0	0.26	0	0.26	0.26	0	0	0.26	SPAM

Summary

- We learned techniques to preprocess textual data
  - Bag of words (tabular structure)
  - There are other ways of imposing a well-defined structure that take grammar into account
    - Word2Vec, FastText, …
    - Beyond the scope of this course
- Next lecture: text mining (part II)