EEP 596: Adv Intro ML || Lecture 5 Recorded (Recorded Lecture)

Dr. Karthik Mohan

Univ. of Washington, Seattle

January 23, 2023

Today's Lecture

1. Feature engineering for Text Data
2. Bag of Words Model
3. TF-IDF

Features for Text Data

→ Spam detection/classification

Email Excerpt

Congratulations! You have been selected for our special offer.

Features for Text Data

Email Excerpt

Congratulations! You have been selected for our special offer.

String to Features

How do we convert strings of text to features that we can use?

Features for Text Data

Email Excerpt

Congratulations! You have been selected for our special offer.

String to Features

How do we convert strings of text to features that we can use?

Bag of words Model

Represent every string in terms of a long vector of words (e.g. every possible word in vocabulary). However, only a few elements of those words are non-zero. So its a 'sparse' vector representation!

Features for Text Data

* Each of the 3 sentences below can be represented as a (multi-hot encoding) bag of words model

Bag of words Example

Sentence 1: This is a simple sentence
Sentence 2: How about this one
Sentence 3: One more

Vocab :- 3 sentences give rise to
V = {'This', 'is', 'a', 'simple', 'sentence', 'How', 'about', 'one', 'more'}
↑ [vocabulary from 3 sentences]

Bag of Words Model | Multi-Hot Encoding

'one'	'This'	'is'	'a'	'simple'	'sentence'	'How'	'about'	'more'
0	1	1	1	1	1	0	0	0
1	1	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1

S1 ←
S2 ←
S3 ←

Vocab Size :- |V| = 9
↑ Cardinality of V

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Example of Pre-Processing

Sentence: Congratulations!!! You have been selected for our special offer.
Sentence after pre-processing: Congratulations have been selected our special offer

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Example

Sentence: Congratulations!!! You have been selected for our special offer.
Sentence after pre-processing: Congratulations have been selected our special offer

After pre-processing

After pre-processing of the sentence, bag of words can be used to vectorize each pre-processed sentence like on the previous slide!

Features for spam classification

Presence or Absence of a word in BOW :- Linear Model

Non-linear features

Some times combination of words can be more useful in making predictions. E.g. "Not bad" may indicate a positive sentiment while just "not" or "bad" indicates negative sentiment. A new feature, "not_bad" can capture this combination and is called a bi-gram.

(Single words) Uni-gram

(2 words Combined) Bi-gram

bad → -ve Sentiment

Not bad → +ve Sentiment!

Bigram

"N-grams"

N=1 → uni-gram

N=2 → bi-gram

Bow Model

Features for spam classification

Non-linear features

Some times combination of words can be more useful in making predictions. E.g. "Not bad" may indicate a positive sentiment while just "not" or "bad" indicates negative sentiment. A new feature, "not_bad" can capture this combination and is called a bi-gram.

N-grams ✓

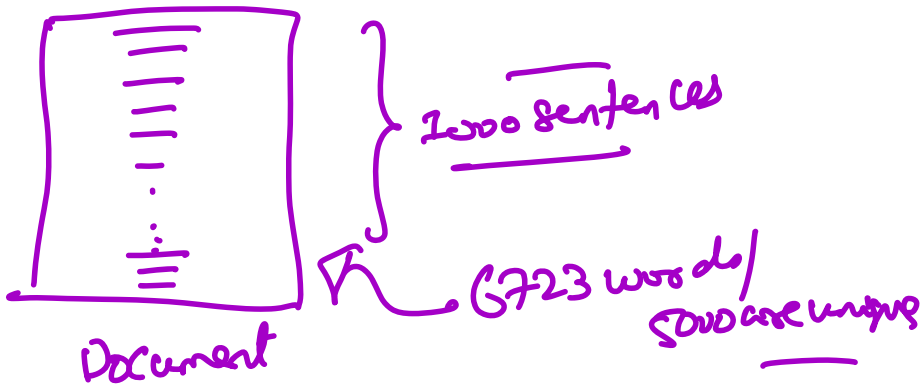
Combination of n-consecutive words can be a feature in the bag of words model. Usually N = 1,2,3 (uni, bi and tri-grams).

ICE #1

Bag of words

Let's say you have 1000 sentences in a document and you want to represent each sentence in the document with bag of words. There are a total of 6723 words in the document with 5000 unique words. What would be the dimension of the vector that represents each sentence using the bag of words model (assume uni-grams and no pre-processing of the sentence)?

a) 5000 ✓
b) 6723
c) 1000
d) 5723



ICE #1

Bag of words

Let's say you have 1000 sentences in a document and you want to represent each sentence in the document with bag of words. There are a total of 6723 words in the document with 5000 unique words. What would be the dimension of the vector that represents each sentence using the bag of words model (assume uni-grams and no pre-processing of the sentence)?

a) 5000
b) 6723
c) 1000
d) 5723

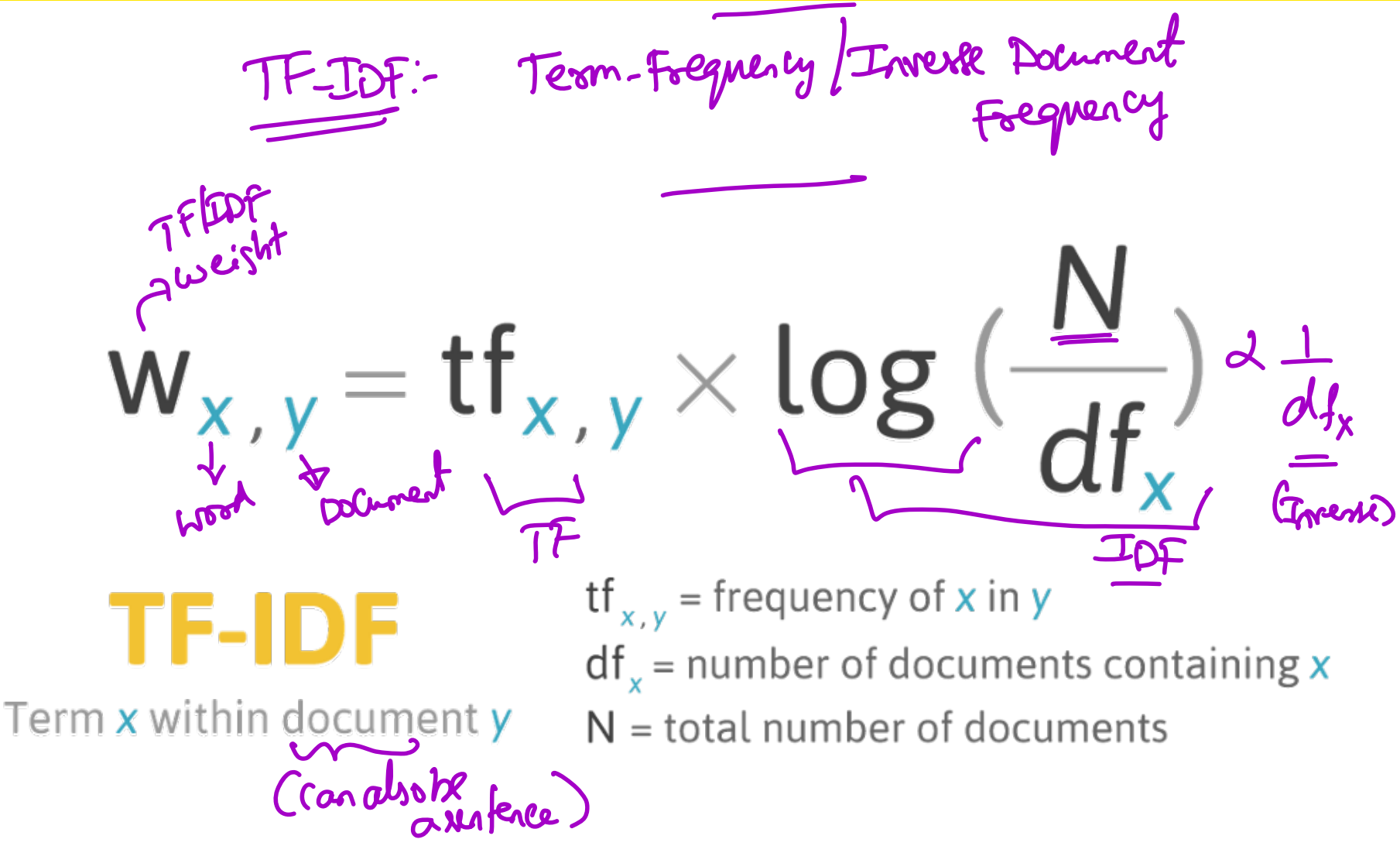
"out of vocab"

New words in the evaluation data set?

Easy Baseline - Ignore OOV words during prediction/inference

How do you deal with words you have not seen in training show up in test?

TF-IDF


TF-IDF:- Term-Frequency / Inverse Document Frequency
TF/IDF weight
$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$
word
document
TF
IDF
$\propto \frac{1}{df_x}$
(Inverse)
TF-IDF
Term x within document y
(can also be sentence)
$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

TF-IDF of you in Email 1

$\text{TF} = 1$

$\text{IDF} = \log(3/2) = \log(1.5)$

$\text{TF-IDF} = \log(1.5)$

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

TF-IDF of you in Email 1

TF-IDF(presentation) > TF-IDF(you)

$\text{TF} = 1$

$\text{IDF} = \log(3/2) = \log(1.5)$

$\text{TF-IDF} = \log(1.5)$

TF-IDF of presentation in Email 3

$\text{TF} = 1$

$\text{IDF} = \log(3) = \log(3)$ $= \log(N/\text{df}(\text{presentation})) = \log(3/1) = \log(3)$

$\text{TF-IDF} = \log(3)$

ICE #2

TF-IDF

Consider the following sentences:

S1: Are you in Seattle right now?

S2: What's the time right now?

S3: Right, that doesn't sound right

If we use a TF-IDF filter, which word is most likely to get eliminated from any of the sentences?

- a. the
- b. you
- c. right
- d. now

Right

TF('Right'/S1) = 1

IDF('Right') = 3

TF-IDF('Right'/S1) = log(N/df) × TF = log(3/3) × 1 = 0

Scalable representations?

"Representation learning" / "Embeddings"
ICLR

Learned feature representations

What if we have 1000's of documents, each with 1000 sentences and an average of 5 words per sentence! That gives us 5 MM words and let's say 300k unique words. Bag of words representation for a sentence becomes memory intensive and also computationally cumbersome!

vocab size, $|V| = 300k$

Sentence Embeddings

$[\,] \in \mathbb{R}^{256}$ → Dense / Compact Rep for a sentence

Enter: Low-dimensional (256 or 512 dimensional) 'learned' features. E.g. Glove Embedding/Word2Vec Embedding. These embeddings are learned from data and are a much more **concise and scalable representation** of a sentence than bag-of-words model could possibly do.

Summary

- Understanding Logistic Regression
- Evaluation metrics for classifiers
- Feature engineering for spam classification

* BoW model
* TF-IDF