Chapter 3

Date ..../..../........

Examples of word Prediction
Google Ngram Viewer
What are you? is high et.

Why probablity?
→ Provides methods to predict or make decision to pick the next word in the sequence based on a sampled data.
→ Make the informed decision when there a certain degree of uncertainity & some observed data.
→ It provides a quantitative description of the chances or likelihood associated with various outcomes.
- Probablity of a sentence
→ probablity of next word in a sentence

⇒ Probablity :- The probablity is defined as the likelihood that an event will occur

Discrete sample space
→ Experiment :- Extracting tokens from a document
→ Outcome :- Every token/ word x in the document
→ The outcome of the experiment - 52 sample (words). They constitute the sample space, Ω or the set of all possible outcomes.
Each word in this sample belongs to Ω represented by $x \in \Omega$

Spiral

Date ..../..../........

Each sample $x \in \Omega$ is assigned a probablity score $[0, 1]$.
A probablity function or probablity distribution function distributes the probablity mass of 1 to the all the samples in the sample space Ω.

⇒ ~~all~~ Sample Space - constraints
All the words in the Ω must satisfy the following contraints:
1. $P(x) \in [0, 1] \quad \forall x \in \Omega$ &
2. $\sum_{x \in \Omega} P(x) = 1$

Example:-
If we are equally likely to pick any word from the BOW, then the probability for any word is
Bag of words Count = 52
~~Ex~~ $P(x) = 1/52 \quad \forall x \in \Omega$ so that
$P(\Omega) = 1$
$P(\text{'weather'}) = 1/52 = 0.01923076923$

⇒ EVENTS
Example:-
Total no. of words = 52
The no. of unique words = 37 or there are 37 types of words have frequencies > 1.
An event is an collection of samples of the same type, $E \subseteq \Omega$

$$P(E) = \sum_{x \in E} P(x)$$

Events can be described as a variable taking a certain value

Spiral

Date ..../..../.........

In the BOW, the word type "the" occurs 6 times. Then

$$E_{the} = 6$$

$$P(E_{the}) = 6 \times \frac{1}{52} = 0.115$$

In the BOW, the word type pack occurs 3 times. Then

$$E_{pack} = 3$$

$$P(E_{pack}) = 3 \times \frac{1}{52} = 0.058$$

Random variable

A random variable, is a variable whose possible values are numerical outcoms of a random phenomenon

→ Two types - continuous and discrete for NLP, they are discrete

To capture the type - token distinction, we random variable W. $W(x)$ maps to the sample $x \in \Omega$.

V is the set of types & the value is represented by a variabe v.

Given a random variable V & a value v, $P(V=v)$ is the probablity of the event that V takes the value v, i.e

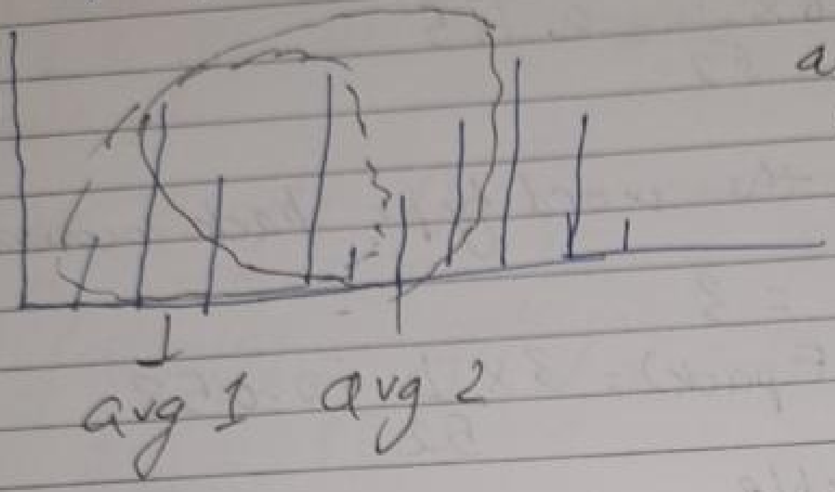
$$P(V=v) = P(x \in \Omega : V(x) = v)$$

$$P(V='the') = P('the') = 0.115$$

Random Variables are useful in describing, constructing various events.

Project - Stock Price Prediction

→ rolling avg - helpful for traders



avg moves fwd 1 at a time

Neural Machine Translation → Hindi → English

→ sequential LSTM

seq → seq

encoder → decoder input → decoder

eng

hindi

we will use sequential model - LSTM

Date ..../..../........

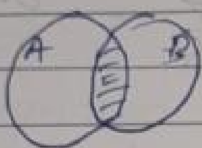
Joint Probablity :-

Given any two events E_1 & E_2, the probablity of their conjuction

$$P(E_1 \cup E_2) = P(E_1 \cap E_2)$$

is called the joint probablity of E_1 & E_2 occurs simultaneously.

Example :- The probablity of the first letter of 't' and the second letter 'h' is $P(F = 't', S = 'h')$. The joint probablity should be as large as the probability of P('the')



$P(A)$ = size of A relative to Ω

$P(A, B)$ = size of $A \cap B$ relative to Ω

Conditional Probablity

When we have partial knowledge influencing the outcome of an experiment, we use it to update the outcome.

The conditional probablity $P(E_2|E_1)$ is the probablity of event E_2 given that event E_1 has occured. $P(E_2|E_1)$ is defined as:

$$P\left(\frac{E_2}{E_1}\right) = \frac{P(E_1 \cap E_2)}{P(E_1)} \quad \text{if } P(E_1) > 0$$

⇒ example - Conditional Probablity - Bigram

let us consider a corpus of Kinematics problems in physics that contains about 280+ problems.

Date ..../..../........

→ Bigram sample space - $\{w_1, w_2\} \in \Omega$: 37A

→ $A(\{w_1, w_2\}) = \{$average, $x\}$ - bigram starting with avg

→ $B(w_1, w_2) = \{x$, speed$\}$ - bigram ending with speed

P(average) = 0.036

P(speed) = 0.114

P(average, speed) = 0.004

P(speed / average) = $\frac{0.004}{0.036}$ = 0.111

P(avg / speed) = $\frac{0.004}{0.114}$ = 0.035

Independence

Two events are dependent if the probablity of one relies on occurrence of the other; if there is not such interaction; then the events are independent.

Two events E_1 & E_2 are independent if & only if $P(E_1, E_2) = P(E_1) P(E_2)$

OR

→ $P(E_1) = P(E_1 | E_2)$ $P(E_2) = P(E_2 | E_1)$

Example :-

1. P(average) = 0.036

P(speed) = 0.114

P(average, speed) = 0.004

The bigram {average, speed} did not happen by chance. The words average, speed are NOT independent

Spiral

The Language Model

→ Natural language sentences can be described as parse trees which use the morphology of words, syntax & semantics.

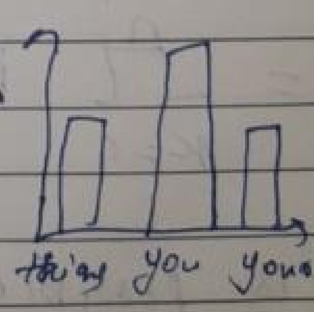
→ Probablistic thinking - finding how likely a sentence occurs or formed, given the word sequence

→ In probablistic world, the language model is used to assign a probability $P(w)$ to every possible word sequence w.

Application ⇒

Speech Recognition	Did you hear Recognize speech or wreck a nice beach.
Context sensitive spelling	Once upon a tie. Their lived a king
Machine Translation	artwork is good → l'oeurve est bonne
Sentence Completion	Complete a sentence as the previous word is given
OCR & Hand writing Recognition	the quick brown fox

predict the next word

How are → Language Model → 

↑ Input Sentence

↑ knowledge about the language - grammar sentence structure, domain etc

Probablistic Language Model

Goal :- compute the probablity of a sequence of words

$P(W) = P(w_1, w_2, w_3 \ldots w_n)$

Task :- To predict the next word using probabitity! Given the content, find the next word using

$$P(w_n | w_1, w_2, w_3 \ldots w_{n-1})$$

A model which computes the probabity for (5) using (6) is called as Probablistic Language Model.

The probability of P(The cat roars) is less likely to happen than P(The cat meows)

Chain Rule

It is difficult to compute the probability of the entire sequence $P(w_1, w_2, w_3, \ldots w_n)$?

Chain rule is used to decompose the joint probablity of a sequence into a product of conditional probablity.

$P(w) = P(w_1, w_2, w_3 \ldots, w_n) = P(w_1^n)$

$= P(w_1) P(w_2 | w_1) P(w_3 | w_2, w_1) \ldots$

$$= \prod_{k=1}^{n} P(w_k | w_1^{k-1})$$

It is possible to $P(w/h)$ but it doesn't really help in reduceing the computational complexity

Spiral

- → We use innovative ways to string words to form new sentences.
- → Finding the probability for a long sentence may not yield good outcome as the content may never occur in the corpus.
- → Short sequences may provide better results.

Markov Assumption

The future behaviour of a dynamic system depends on its recent & not on the entire history.

The product of the conditional probablities can be written approximatly for a bigram as

$$P(w_k / w_1^{k-1}) \approx P(w_k / w_{k-1}) \quad — (10)$$

equation (10) can be generalized for an n-gram as

$$P(w_k / w_1^{k-1}) \approx P(w_k / w_{k-K+1}^{k-1})$$

Now, the joint probability of a sequence can be re-written as

$$P(w) = P(w_1, w_2, w_3 \ldots w_n) = P(w_1^n)$$

$$= P(w_1)\, P(w_2/w_1)\, P(w_3 / w_2 w_1) \ldots$$

$$= \prod_{k=1}^{n} (w_k / w_1^{k-1})$$

$$\approx \prod_{k=1}^{n} P(w_k / \underbrace{w_{k-K+1}^{k-1}}_{n\text{-gram}})$$

Date ..../..../.........

Generative Models

⇒ Target & Context words
Next words in the sentence depends on its immediate past words, known as context words

$$P(\omega_{K+1} \mid \underbrace{\omega_{i-K}, \omega_{i-K+1} \ldots \omega_K}_{\text{content words}})$$

n-grams
unigram − $P(\omega_{K+1})$
bigram − $P(\omega_{K+1} \mid \omega_K)$
trigram − $P(\omega_{K+1} \mid \omega_{K-1}, \omega_K)$
4-gram − $P(\omega_{K+1} \mid \omega_{K-2}, \omega_{K-1}, \omega_K)$

Language Modelling using Unigrams

→ A unigram language model all words are generated independently $\omega_1, \omega_2, \omega_3, \ldots, \omega_n$ and none of them depend on the other.
→ This is not a good model for language generation
→ It may generate the the the the as a sentence
→ cannot string words according to high probablity

→ Generates a document containing N words using n-gram.
→ A good model assigns higher probablity to the word that actually occurs

→ $\sum_{i=1}^{N} P(\omega_i) = 1$ ω_i to be estimated

Spiral

Date ..../..../........

in this model is $P(w_i)$ & it must satisfy this

$$P(w) = \prod_{i=1}^{N} P(w_i)$$

→ The location of the word in the document is not important

→ $P(N)$ is the distribution over N & is same for all documents

Maximum Likelihood Estimate

→ One of the methods to find the unknown parameter(s) is the use of Maximum Likelihood Estimate.

→ Estimate the parameter value for which the observed data has the highest probablity.

→ Training data may not have all the words in a vocabulary.

→ If a sentence with an unknown word is presented, then the MLE is 0.

→ Add a smoothing parameter to the equation without affecting the overall probablity requirements

$$P(w) = \frac{C_{w_i} + \alpha}{C_w + \alpha |W|}$$

Bigram Language Model

A bigram language model generates a sequence one word at a time, starting with the first word & then generating each succeeding word conditioned on the previous one.

Spiral

Date ..../..../.........

→ A bigram model is defined as follows :-

$$P(w) = \prod_{i=1}^{n+1} P(w_i / w_{i-1}),$$

where $W = w_1, w_2 \ldots w_n$

→ Estimate the parameter $P(w_i / w_{i-1})$ for all bigrams.

→ The parameter estimation does not depend on the location of the word.

→ If we consider the sentence as a sequence in time, they are time-invariant MLE picks up the word that is $\frac{n_{w,w'}}{n_{w,0}}$

where $n_{w,w'}$ is the number of times the words w, w' occur together & $n_{w,0}$ is the number of times the word w appears in the bigram sequence.

Probablistic Language Model - Example

1] Peter Piper picked a peck of pickled pepper.
2] A peck of pickled peppers Peter Piper picked.
3] If Peter Piper picked a peck of pickled peppers
4] Where's the peck of pickled peppers Peter Piper picked?

Bigram	Freq
<S> Peter	1
Peter Piper	4
Piper picked	4
picked a	2
a peck	2
peck of	4
:	

Spiral

Date ..../..../........

The joint probability of a sentence formed with n words can be expressed as a product of conditional probabilities - we use immediate context & not the entire history.

$P(w_1 | <S>) \times P(w_2 / w_1) \times \ldots P(<E> / w_n)$

and $P(w_{i+1} / w_i) = \frac{|(w_i \cdot w_{i+1})|}{|w_i|}$

Out of Vocabulary Words

→ In a closed vocabulary language model, there is no unknown words or out of vocabulary words (OOV)

→ In an open vocabulary system, you will find new words that are not present in the trained model.

→ Pick words below certain frequency and replace them as OOV.

→ Treat every OOV as a regular word.

→ During testing the new words would be treated as OOV & the corresponding frequency will be used for computation

→ this eliminates zero probability for sentences containing OOV

Curse of Dimensionality

→ A fundamental problem that makes language and other learning problems difficult is the curse of dimensionality

→ It is particularly obvious in the case when one wants to model the joint

Spiral

Date ..../..../........

distribution b/w many discrete random variable
→ If one wants to estimate the joint probability distribution of 10 words in a language with a million words as vocabulary, then we need to estimate $1000000^{10} - 1 = 10^{60} - 1$ free parameters.

Naive Bayes Classification

⇒ Bayes Theorem
Let us consider two random variables X & Y. Then joint probability. Then joint probability $P(X=x, Y=y)$ refers to the probability that the variable X takes the value x and the variable Y takes the value y. The conditional probability $(P(Y=y \mid X=x)$ refers to the probability that the variable Y takes the value y given the observation the variable X takes the value x.

$$P(X,Y) = P(Y/X) \times P(x) = P(X/Y) \times P(Y)$$

$$P(Y/X) = \frac{P(X/Y)\, P(Y)}{P(X)}$$

Bayes theorem for Email Classification
→ Map Baye's theorem using statistical properties of data
→ Let $X = \{X_1, X_2, X_3 \ldots X_n\}$ where X is a set of attributes & Y represents a class
→ The relationship b/w X & Y can be found using the conditional properties $P(Y/X)$

Spiral

→ The conditional probability $P(y|X)$ is known as the posterior probability of y
→ $P(y)$ is known as the prior probability
→ In the classification problem, it is important to learn the parameters $P(y|X)$. Given the attributes of the email (TF, TF-IDF) find the class to which the email belongs
→ The parameters are obtained from training data. During the training process, we will learn $P(y|X)$ for every word in the corpus.

Supervised Classification.
→ Set of input parameters / attributes $X = x_1, x_2 \ldots x_n$ and a fixed set of classes $Y = y_1, y_2 \ldots y_n$
→ Every element of the training set $D = d_1, d_2 \ldots d_n$ is manually assigned a class
(d_1, y_1) (d_2, y_2) .. (d_n, y_n).
→ Goal is to learn the classifier, so that it can map a new document d' to any of the classes $y \in Y$
→ Bayes classifier would assign a probability based on the observation to the new document to aid the class selection.
→ The probability score for each class is computed as given by the the equation

$$P(y|x) = \frac{P(X|y)\,P(y)}{P(X)}$$

→ The class will be found using argmax $P(y|X)$

Date ..../..../.........

$$\hat{y} = \underset{y \in Y}{\arg\max}\ P(y/X)$$

$$= \underset{y \in Y}{\arg\max}\ P(X/y)\ P(y)$$

$$= \arg\max\ P(y)\ P(X_1/y) \times P(X_2/y) \times \ldots P(X_m/y)$$

$$= \underset{y \in Y}{\arg\max}\ P(y) \prod_{i=1}^{m} P(X_i/y)$$

TRAINING

1] Prior Probability $P(y) = \dfrac{Count(y)}{Count(Y)} = 1/2$

2] Learn $P(X_1/y) = \dfrac{Count(X_1, y)}{Count(y)}$

given the class,
find the probability of
the word in it

for new word, the probability will become 0
so we add 1 to all numerators
dont change denominator. — Smoothing
every corpus is not complete hence
smoothing is important acc to domain.