Vijay Choubey

Summary

The website content provides an in-depth explanation of Non-Negative

Topic Modelling Using NMF

Well, In this blog I want to explain one of the most important concept of Natural Language Processing. I'm excited to start with the concept of Topic Modelling. So lets first understand it.

Dynamic Topic Modeling

concept of the traditional Natural Processing Approach because of its potential to obtain semantic relationship between words in the document clusters. In addition that, it has numerous other applications in NLP.

Some of the well known approaches to perform topic modeling are

- Non-Negative Matrix Factorization (NMF)
- Latent Dirichlet Allocation (LDA)
- Latent Semantic Analysis (LSA)

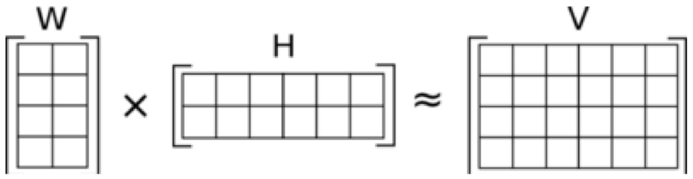
Now let us have a look at the Non-Negative Matrix Factorization.

Non-Negative Matrix Factorization

Non-Negative Matrix Factorization is a statistical method to reduce the dimension of the input corpora. It uses factor analysis method to provide comparatively less weightage to the words with less coherence.

For a general case, consider we have an input matrix V of shape *m x n*. This method factorizes V into two matrices W and H, such that the dimension of W is *m x k* and that of H is *n x k*. For our situation, V represent the term document matrix, each row of matrix H is a word embedding and each column of the matrix W represent the weightage of each word get in each

But the assumption here is that all the entries of W and H is positive given that all the entries of V is positive.



Now let us look at the mechanism in our case. Suppose we have a dataset consisting of reviews of superhero movies. In the document term matrix (input matrix), we have individual documents along the rows of the matrix and each unique term along the columns. In case, the review consists of texts like Tony Stark, Ironman, Mark 42 among others. It may be grouped under the topic **Ironman**. In this method, each of the individual words in the document term matrix are taken into account. While factorizing, each of the words are given a weightage based on the semantic relationship between the words. But the one with highest weight is considered as the topic for a set of words. So this process is a weighted sum of different words present in the documents.

Math behind NMF

the elements. The distance can be measured by various methods. Some of them are Generalized Kullback–Leibler divergence, frobenius norm etc.

1. Generalized Kullback–Leibler divergence

It is a statistical measure which is used to quantify how one distribution is different from another. Closer the value of Kullback–Leibler divergence to zero, the closeness of the corresponding words increases. In other words, the divergence value is less.

Let us look at the difficult way of measuring Kullback–Leibler divergence. Formula for calculating the divergence is given by.

Python Implementation of the formula is shown below.

```
from numpy import sum
def kullback_leibler_divergence(p, q):
    return sum(p[i] * log2(p[i]/q[i]) for i in range(len(p)))

a=[0.78, 0.25, 0.98, 0.35]
b=[0.58, 0.46, 0.28, 0.17]
kullback_leibler_divergence(a, b)
```

```
from scipy.special import kl_div
a=[0.78, 0.25, 0.98, 0.35]
b=[0.58, 0.46, 0.28, 0.17]
print(kl_div(a,b))
```

2. Frobenius Norm

The other method of performing NMF is by using Frobenius norm. It is defined by the square root of sum of absolute squares of its elements. It is also known as eucledian norm. The formula and its python implementation is given below.

```
import numpy as np
a=[0.78, 0.25, 0.98, 0.35]
frobenius_norm = numpy.linalg.norm(a)
```

An optimization process is mandatory to improve the model and achieve high accuracy in finding relation between the topics. There are two types of optimization algorithms present along with scikit-learn package.

- Multiplicative update Solver

Build better voice apps. Get more articles & interviews from voice technology experts at voicetechpodcast.com

NMF in action

Why should we hard code everything from scratch, when there is an easy way? Packages are updated daily for many proven algorithms and concepts. We have a scikit-learn package to do NMF. We will use the 20 News Group dataset from scikit-learn datasets. We will first import all the required packages.

```
# Importing Necessary packages

import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF
```

Now let us import the data and take a look at the first three news articles.

```
text_data= fetch_20newsgroups(remove=('headers', 'footers', 'quotes')).data
text_data[:3]
```

Now, we will convert the document into a term-document matrix which is a collection of all the words in the given document.

```
# converting the given text term-document matrix

vectorizer = TfidfVectorizer(max_features=1500, min_df=10, stop_words='englis
X = vectorizer.fit_transform(text_data)
words = np.array(vectorizer.get_feature_names())

print(X)
print("X = ", words)
```

Defining term document matrix is out of the scope of this article. In brief, the algorithm splits each term in the document and assigns weightage to each words.

Now, let us apply NMF to our data and view the topics generated. For ease of understanding, we will look at 10 topics that the model has generated. We will use Multiplicative Update solver for optimizing the model.

```
nmf = NMF(n_components=10, solver="mu")
W = nmf.fit_transform(X)
H = nmf.components_

for i, topic in enumerate(H):
     print("Topic {}: {}".format(i + 1, ",".join([str(x) for x in words[topic
```

Now the topics are listed below.

> *Topic 1: really,people,ve,time,good,know,think,like,just,don Topic 2: info,help,looking,card,hi,know,advance,mail,does,thanks Topic 3: church,does,christians,christian,faith,believe,christ,bible,jesus,god Topic 4: league,win,hockey,play,players,season,year,games,team,game Topic 5: bus,floppy,card,controller,ide,hard,drives,disk,scsi,drive Topic 6: 20,price,condition,shipping,offer,space,10,sale,new,00 Topic 7: problem,running,using,use,program,files,window,dos,file,windows Topic 8: law,use,algorithm,escrow,government,keys,clipper,encryption,chip,key Topic 9: state,war,turkish,armenians,government,armenian,jews,israeli,israel,pe ople Topic 10: email,internet,pub,article,ftp,com,university,cs,soon,edu*

For crystal clear and intuitive understanding, look at the topic 3 or 4. In topic 4, all the words such as “league”, “win”, “hockey” etc. are related to sports and

The Factorized matrices thus obtained is shown below.

W matrix can be printed as shown below. For the sake of this article, let us explore only a part of the matrix.

```
print(W[:10,:10])
```

H matrix is given below.

```
print(H[:10,:10])
```

It is quite easy to understand that all the entries of both the matrices are only positive.

Don’t trust me? Go on and try hands on yourself. By following this article, you can have an in-depth knowledge of the working of NMF and also its practical implementation. If you have any doubts, post it in the comments.

When can we use this approach?

the entries are close to zero and only very few parameters have significant values. This can be used when we strictly require fewer topics.

- NMF produces more coherent topics compared to LDA.

I will be explaining the other methods of Topic Modelling in my upcoming articles.

Thanks for reading!.I am going to be writing more **NLP articles** in the future too. **Follow** me up to be informed about them. And I am also a freelancer,If there is some freelancing work on data-related projects feel free to reach out over **Linkedin**.Nothing beats working on real projects!

Clap if you liked the article!

Something just for you

Hi! I'm Carl, host of the Voice Tech Podcast

So you're interested in voice technology?

Yes No

Just now

Naturallanguageprocessing Signal Processing Speech Recognition Audio Conversational UI

Recommended from ReadMedium