Vector Space Model Uygulama Raporu

Hazırlayan

Fırat Kaan Bitmez

Öğrenci Numarası

23281855

Dersin Hocası

Asst.Prof.Dr. İsmail İşeri

Giriş

Bu Raporda Vektör Uzayı Modelleri, vektörlerle temsil edilen veriler arasındaki ilişkiyi dikkate alacaktır. Bilgi erişim sistemlerinde popüler ancak başka amaçlar içinde faydalıdır. Genel olarak iki vektörün benzerliği geometrik açıdan karşılaştırmamıza olanak tanır.

Amacımız

Bu çalışmanın amacı, Vector Space modelinin ne olduğunu öğrenmek ve uygulama yaparak neler yapabileceğini görmek.

Bu rapor Sonucunda elde edeceğimiz öğrenme çıktıları şunlar olacaktır:

- Vector Space modeli nedir ve kosinüs benzerliğinin özellikleri?
- Kosinüs benzerliği iki vektörü karılaştırmamıza nasıl yardımcı olabilir?
- Kosinüs benzerliği ile L2 mesafesi arasındaki fark nedir?

Kütüphaneler, Araçlar ve Modüller

Python programlama dili projeyi gerçekleştirmek için kullanıyoruz

Pandas_datareader Modülü veri okuma için kullanıyoruz (**pip install pandas_datareader)**

Distutils: Python'un standart kütüphanesinde bulunan ve dağıtılabilir Python paketlerinin oluşturulması, dağıtılması ve yüklenmesini kolaylaştıran bir araçtır. **(pip install distutils)**

Pandas_datareader : Güncelleme Komutu (**pip install --upgrade pandas_datareader)**

Setuptools: Python'da paket oluşturmayı, dağıtmayı ve yüklemeyi kolaylaştıran bir kütüphanedir. (**pip install setuptools)**

Vector Space ve Kosinüs Formülü

Vektör uzayını dikkate almak faydalıdır çünkü bir şeyleri vektör olarak temsil etmek faydalıdır. Örneğin makine öğreniminde genellikle birden fazla özelliğe sahip bir veri noktamız vardır. Bu nedenle uygun bir veri noktasını vektör olarak temsil etmemizi sağlar.

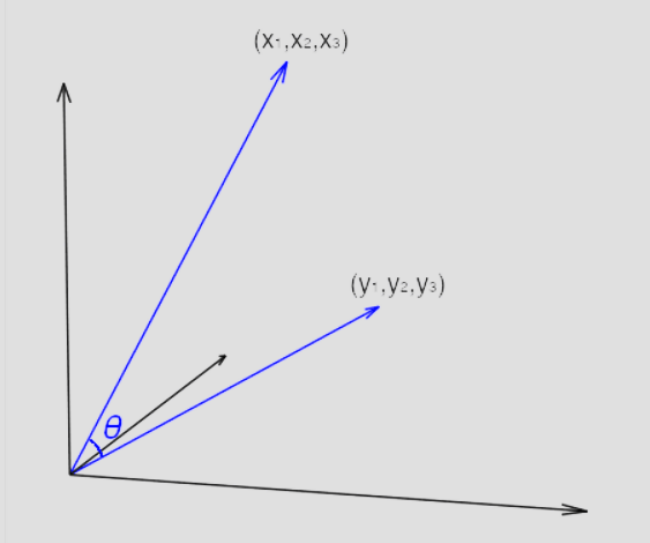
Bir vektör kullanarak normunu(uzaklığını) hesaplayabiliriz. En yaygın olan **L2** normu veya uzunluğudur.Aynı vektör uzayındaki iki vektörle aralarındaki farklı bulabiliriz. **Örneğin 3 boyutlu bir vektör uzayı olduğunu varsalım**: iki vektör (x1,x2,x3) ve (y1,y2,y3) olsun.

Bu vektörler aralarındaki farklar(mesafeler) ise (y1-x1, y2-x2,y3-x3) ve **L2 Normu** bu iki vektör arasındaki mesafe daha doğrusu **öklit** uzaklığıdır.

$$\sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + (y_3 - x_3)^2}$$

Mesafenin yanı sıra iki vektör arasındaki **Açıyı** dikkate alabiliriz

Üç boyutlu koordinat sisteminde (x1,x2,x3) vektörü (0,0,0) noktasından (x1,x2,x3) noktasına bir doğru parçası olarak düşünürsek (0,0,0) ile (y1,y2,y3). Kesişme noktarında bir açı yaparlar:



İki çizgi arasındaki açı kosinüs formülü kullanılarak bulunabilir.

$$\cos\theta = \frac{a \cdot b}{\|a\|_2 \|b\|_2}$$

Benzerlik için Vector Space Modelini Kullanma

Vektör uzay modelinin ne kadar kullanışlı olduğuna dair bir örneğe bakalım.

İlk olarak Veri okuma için **Pytondaki pandas_datareader** modülünü yüklemeliyiz

Bunun için Terminale "**pip install pandas_datareader**" komutunu yazıyoruz.

Dünya Bankası tarafından toplanan veri serileri bir tanımlayıcıyla adlandırılmaktadır. Örneğin "SP.URB.TOTL" bir ülkenin toplam kentsel nüfusudur. Dizilerin çoğu yıllık. Bir diziyi indirdiğimizde, başlangıç ve bitiş yıllarını girmeniz gerekir. Genellikle veriler zamanında güncellenmez. Bu nedenle, eksik veriyi önlemek için en son yıl yerine birkaç yıl geriye dönük verilere bakmak en iyisidir. Aşağıda, her ülkenin 2010 yılındaki bazı ekonomik verilerini toplamaya çalışıyoruz: Yukarıda her ülkenin bazı ekonomik ölçümlerini elde ettik.

Bu Verileri elde Etmek için kullandığımız Kodlamamız

```
from pandas_datareader import wb
import pandas as pd
pd.options.display.width = 0

names = [
    "NE.EXP.GNFS.CD", # Exports of goods and services (current US$)
    "NE.IMP.GNFS.CD", # Imports of goods and services (current US$)
    "NV.AGR.TOTL.CD", # Agriculture, forestry, and fishing, value added (current
US$)
    "NY.GDP.MKTP.CD", # GDP (current US$)
    "NE.RSB.GNFS.CD", # External balance on goods and services (current US$)
]

df = wb.download(country="all", indicator=names, start=2010,
end=2010).reset_index()
countries = wb.get_countries()
non_aggregates = countries[countries["region"] != "Aggregates"].name
df_nonagg = df[df["country"].isin(non_aggregates)].dropna()
print(df_nonagg)
```

Wb.download() kodunun işlevi, Dünya bankasından verileri indirecek ve bir panda veri çerçevesinde döndürecektir.

Wb.get_Countries() Dünya bankası tarafından tanımlanan ülke ve bölgerin adını alacaktır.Bunu Doğu Asya gibi ülke dışı toplamları filtrelemek için kullandık.

Çeşitli nedenlerden dolayı her ülkenin tüm verileri olmayabilir. Bu nedenle eksik veriye sahip olanları kaldırmak için **dropna()** işlevini kullanıyoruz.

Kodlama Debug etmek istedimizde bazı hatalar ile karşılaştık bu hataları çözmek için ise bazı yüklememiz gereken modüller vardı onları yükleyerek çözüme ulaştık.

No module named 'distutils' hatası ile karşılaştık çözüm içinse:

pip install distutils

Diğer hatalar içinse:

pip install --upgrade pandas_datareader

pip install setuptools

Komutlarını terminal üzerinden girerek hatayı giderdik.

Kodlarını Terminal üzerinden girdiğimizde Sorunumuzu çözdük ve Kodlama sonucunda Şöyle bir çıktı elde Ettik:

```
ers\FIRAT\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '55872' '--' 'C:\
ce-model\country_economic_data.py'
C:\Users\FIRAT\Desktop\myProject\NLP-natural-language-processing\Vector-space-model\country_economic_data.py:13: FutureWarning: errors='
 without passing `errors` and catch exceptions explicitly instead
  df = wb.download(country="all", indicator=names, start=2010, end=2010).reset_index()
               country  year  NE.EXP.GNFS.CD  NE.IMP.GNFS.CD  NV.AGR.TOTL.CD  NY.GDP.MKTP.CD  NE.RSB.GNFS.CD
3              Albania  2010    3.337087e+09    5.792187e+09    2.141583e+09    1.192693e+10   -2.455101e+09
4              Algeria  2010    6.197542e+10    5.065474e+10    1.364853e+10    1.612073e+11    1.132068e+10
7               Angola  2010    5.157281e+10    3.568225e+10    5.179053e+09    8.379947e+10    1.589056e+10
10           Argentina  2010    8.020887e+10    6.793793e+10    3.021382e+10    4.236274e+11    1.227093e+10
12               Aruba  2010    1.477512e+09    1.846245e+09    5.027933e+05    2.453597e+09   -3.687335e+08
..                 ...   ...             ...             ...             ...             ...             ...
259          Viet Nam  2010    7.974748e+10    8.802901e+10    2.263231e+10    1.472012e+11   -8.281534e+09
261  West Bank and Gaza  2010    1.367300e+09    5.264300e+09    8.716000e+08    9.681500e+09   -3.897000e+09
263        Yemen, Rep.  2010    9.270503e+09    1.062900e+10    3.051643e+09    3.090675e+10   -1.358501e+09
264              Zambia  2010    7.503513e+09    6.256989e+09    1.909207e+09    2.026556e+10    1.246524e+09
265            Zimbabwe  2010    3.569254e+09    6.440274e+09    1.157187e+09    1.204166e+10   -2.871020e+09

[173 rows x 7 columns]
PS C:\Users\FIRAT\Desktop\myProject>
```

Yukardaki terminal çıktısında gördüğünüz gibi Her ülkenin 2010 yılındaki bazı ekonomik ölçümlerini elde ettik.

Pandas veya numpy fonksiyonundaki gerçek manipülasyonu gizlemek yerine fikri daha iyi açıklamak için, önce her ülkeye ait verileri bir vektör olarak çıkarıyoruz:

```
vectors = {}
for rowid, row in df_nonagg.iterrows():
    vectors[row["country"]] = row[names].values

print(vectors)
```

Bunun sonucunda şöyle bir çıktı ile karşılaşıyoruz:

```
[173 rows x 7 columns]
{'Albania': array([3337086708.7743, 5792187274.96813, 2141583008.42955,
       11926926615.8015, -2455100566.19383], dtype=object), 'Algeria': array([61975419481.9204, 50654743649.7546, 13648525690.6533,
       161207307027.185, 11320675832.1659], dtype=object), 'Angola': array([51572807247.0924, 35682251201.2096, 5179053428.22103,
       83799473759.7201, 15890556045.8828], dtype=object), 'Argentina': array([80208867995.7171, 67937933972.3653, 30213817111.0998,
       423627422092.49, 12270934023.3519], dtype=object), 'Aruba': array([1477511731.84358, 1846245251.39665, 502793.296089385,
       2453597206.70391, -368733519.553073], dtype=object), 'Australia': array([227427026265.26, 237995666496.38, 25325453168.2139,
       1148890200292.42, -10568640231.1202], dtype=object), 'Austria': array([201088694407.167, 187341317488.667, 4971410985.66668,
       392275107258.667, 13747376918.5], dtype=object), 'Azerbaijan': array([28732245203.09, 10942312484.4256, 2921255918.26564,
       52909294791.9262, 17789932718.6643], dtype=object), 'Bahamas, The': array([3528800000.0, 4414100000.0, 110800000.0, 10095760000.0,
       -885300000.0], dtype=object), 'Bahrain': array([17880319148.9362, 13097074468.0851, 76542553.1914894,
       25713271276.5957, 4783244680.85106], dtype=object), 'Bangladesh': array([18472449276.0536, 25106319010.5127, 19598804913.9026,
       115279077465.226, -6633869734.45904], dtype=object), 'Barbados': array([1763300000.0, 2174250000.0, 60050000.0, 4531150000.0, -410950000.0],
      dtype=object), 'Belarus': array([29401680706.1249, 36940382943.1494, 5088786003.74013,
       57231904542.8755, -7538702237.02455], dtype=object), 'Belgium': array([365158050100.001, 356962071951.667, 3683371186.66667,
       481420882905.001, 8195978148.33335], dtype=object), 'Belize': array([885429215.68255, 773628997.521695, 152967107.955376,
       1739070295.44027, 111800218.160855], dtype=object), 'Benin': array([2199999385.40112, 2704110555.34836, 2464041425.56528,
```

Ve bu şekilde devam ediyor. Oluşturduğumuz Python sözlüğü, anahtar olarak her ülkenin adını ve numpy dizisi olarak ekonomik ölçümleri içeriyor. 5 metrik vardır, dolayısıyla her biri 5 boyutlu bir vektördür.

Bunun bize yardımcı olduğu şey, her ülkenin diğerine ne kadar benzer olduğunu görmek için vektör temsilini kullanabilmemizdir. Hem farkın L2 normunu (Öklid mesafesi) hem de kosinüs mesafesini deneyelim. Avustralya gibi bir ülkeyi seçiyoruz ve seçilen ekonomik ölçütlere göre onu listedeki diğer tüm ülkelerle karşılaştırıyoruz.

```
euclid = {}
cosine = {}
target = "Australia"

for country in vectors:
    vecA = vectors[target]
    vecB = vectors[country]
    dist = np.linalg.norm(vecA - vecB)
    cos = (vecA @ vecB) / (np.linalg.norm(vecA) * np.linalg.norm(vecB))
    euclid[country] = dist    # Euclidean distance
    cosine[country] = 1-cos   # cosine distance
```

Yukarıdaki for döngüsünde, vecA'yı hedef ülkenin (yani Avustralya) vektörü olarak ve vecB'yi diğer ülkenin vektörü olarak ayarladık. Daha sonra farklarının L2-normunu iki vektör arasındaki Öklid mesafesi olarak hesaplıyoruz. Ayrıca formülü kullanarak kosinüs benzerliğini hesaplıyoruz ve bunu 1'den çıkararak kosinüs mesafesini elde ediyoruz. Yüzden fazla ülke arasından hangisinin Avustralya'ya en kısa Öklid mesafesine sahip olduğunu görebiliriz:

```
df_distance = pd.DataFrame({"euclid": euclid, "cos": cosine})
print(df_distance.sort_values(by="euclid").head())
```

```
                euclid          cos
Australia  0.000000e+00  2.220446e-16
Mexico     1.414011e+11  6.968702e-03
Spain      3.406184e+11  3.129546e-03
Turkiye    3.825502e+11  3.583513e-03
Indonesia  4.107814e+11  7.402548e-03
PS C:\Users\FIRAT\Desktop\myProject>
```

Sonucu sıraladığımızda Öklid uzaklığı altında Avustralya'ya en yakın olanın Meksika olduğunu görebiliriz.

Ancak kosinüs mesafesiyle Avustralya'ya en yakın yer Kolombiya'dır.

```
df_distance = pd.DataFrame({"euclid": euclid, "cos": cosine})
print(df_distance.sort_values(by="cos").head())
```

```
                    euclid          cos
Australia     0.000000e+00  2.220446e-16
Colombia      9.007020e+11  1.689471e-03
South Africa  7.538465e+11  2.302299e-03
Cuba          1.133521e+12  2.393143e-03
Italy         1.088273e+12  2.756956e-03
PS C:\Users\FIRAT\Desktop\myProject>
```

İki mesafenin neden farklı sonuçlar verdiğini anlamak için üç ülkenin metriğinin birbirleriyle nasıl karşılaştırıldığını gözlemleyebiliriz:

```
print(df_nonagg[df_nonagg.country.isin(["Mexico", "Colombia", "Australia"])])
```

```
       country  year  NE.EXP.GNFS.CD  NE.IMP.GNFS.CD  NV.AGR.TOTL.CD  NY.GDP.MKTP.CD  NE.RSB.GNFS.CD
13   Australia  2010    2.274270e+11    2.379957e+11    2.532545e+10    1.148890e+12   -1.056864e+10
47    Colombia  2010    4.681628e+10    5.135131e+10    1.812061e+10    2.864985e+11   -4.535024e+09
157     Mexico  2010    3.207660e+11    3.344563e+11    3.417981e+10    1.105424e+12   -1.369028e+10
PS C:\Users\FIRAT\Desktop\myProject>
```

Bu tablodan Avustralya ve Meksika metriklerinin büyüklük olarak birbirine çok yakın olduğunu görüyoruz. Ancak aynı ülke içindeki her metriğin oranını karşılaştırırsanız, Avustralya'ya en iyi uyanın Kolombiya olduğunu görürsünüz.

Aslında kosinüs formülünden şunu görebiliriz:

$$\cos\theta = \frac{a \cdot b}{\|a\|_2 \|b\|_2} = \frac{a}{\|a\|_2} \cdot \frac{b}{\|b\|_2}$$

bu, iki vektör arasındaki açının kosinüsünün, karşılık gelen vektörlerin 1 uzunluğa normalize edildikten sonraki nokta çarpımı olduğu anlamına gelir. Dolayısıyla kosinüs mesafesi, mesafeyi hesaplamadan önce verilere sanal olarak bir ölçekleyici uygulamaktır.

Kodlamanın Tamamı

```
from pandas_datareader import wb
import pandas as pd
import numpy as np

pd.options.display.width = 0

names = [
    "NE.EXP.GNFS.CD", # Exports of goods and services (current US$)
    "NE.IMP.GNFS.CD", # Imports of goods and services (current US$)
```

```
    "NV.AGR.TOTL.CD", # Agriculture, forestry, and fishing, value added (current
US$)
    "NY.GDP.MKTP.CD", # GDP (current US$)
    "NE.RSB.GNFS.CD", # External balance on goods and services (current US$)
]

df = wb.download(country="all", indicator=names, start=2010,
end=2010).reset_index()
countries = wb.get_countries()
non_aggregates = countries[countries["region"] != "Aggregates"].name
df_nonagg = df[df["country"].isin(non_aggregates)].dropna()
print(df_nonagg)


print("#########################################")
vectors = {}
for rowid, row in df_nonagg.iterrows():
    vectors[row["country"]] = row[names].values

print(vectors)

print("#########################################")


euclid = {}
cosine = {}
target = "Australia"

for country in vectors:
    vecA = vectors[target]
    vecB = vectors[country]
    dist = np.linalg.norm(vecA - vecB)
    cos = (vecA @ vecB) / (np.linalg.norm(vecA) * np.linalg.norm(vecB))
    euclid[country] = dist    # Euclidean distance
    cosine[country] = 1-cos   # cosine distance

print("#########################################")

df_distance = pd.DataFrame({"euclid": euclid, "cos": cosine})
print(df_distance.sort_values(by="euclid").head())

print("#########################################")

df_distance = pd.DataFrame({"euclid": euclid, "cos": cosine})
print(df_distance.sort_values(by="cos").head())
```

```
print("#########################################")

print(df_nonagg[df_nonagg.country.isin(["Mexico", "Colombia", "Australia"])])
```

Sonuç

Sonuç olarak Space Vector Hakkında genel bilgi sahibi olundu.Bir Uzay Vektörün nasıl oluşturulduğu ve oluşturuken nelere ihtiyacımızın olduğunu uygulamalı olarak öğrenmiş olduk. Uzay vektör ile kosinüs arasındaki benzerliği uygulamalı olarak görüp test ettik. Ve aralarındaki Öklid mesafesi gibi ölçümleri nasıl yorumlayacağımızı görmüş olduk.

Kaynaklar

https://machinelearningmastery.com/a-gentle-introduction-to-vector-space-models/