Physics Dataset Practice Problems

Week of July 27, 2020

3 Medical Physics Abstracts

3.1 Data Import

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.close()

#read dataset
with open('medphys.nbib','r') as filein:
    dataset = filein.readlines()

#parsing dataset into title and abstract list
titles = []
abstracts = []
paper_number = 0
intitle = 0
inabstract = 0

for a in range(len(dataset)):
    #counts number of papers
    if (dataset[a][0:5] == 'PMID-'):
        #checks to make sure you only add when both abstracts and titles present
        paper_number = min(len(abstracts),len(titles))
        abstracts = abstracts[0:paper_number]
        titles = titles[0:paper_number]
        paper_number += 1
    #builds title list
    if intitle and not(dataset[a][0:5] == '     '):
        intitle = 0
    if (dataset[a][0:5] == 'TI  -'):
        titles.append(dataset[a][5:-1])
        intitle = 1
    if intitle and (dataset[a][0:5] == '     '):
        titles[paper_number - 1] += dataset[a][5:-1]
    #builds abstract list
    if inabstract and not(dataset[a][0:5] == '     '):
        inabstract = 0
    if (dataset[a][0:5] == 'AB  -'):
        abstracts.append(dataset[a][5:-1])
        inabstract = 1
    if inabstract and (dataset[a][0:5] == '     '):
        abstracts[paper_number - 1] += dataset[a][5:-1]

paper_number = min(len(abstracts),len(titles))
abstracts = abstracts[0:paper_number]
titles = titles[0:paper_number]
```

3.2 Preprocessing

```
import re, nltk
#nltk.download('stopwords') #don't need to do this each time
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer #stemming only looks at roots of words
ps = PorterStemmer()
all_stopwords = set(stopwords.words('english'))
#all_stopwords.remove('not')
for a in range(len(titles)):
    #get rid of non letters, lower case, and split into separate words
    titles[a] = re.sub('[^a-zA-Z]' , ' ' , titles[a]).lower().split()
    abstracts[a] = re.sub('[^a-zA-Z]' , ' ' , abstracts[a]).lower().split()
    #remove all stop words
    titles[a] = [ps.stem(word) for word in titles[a] if not word in all_stopwords]
    abstracts[a] = [ps.stem(word) for word in abstracts[a] if not word in all_stopwords]
    #join words back together
    titles[a] = ' '.join(titles[a])
    abstracts[a] = ' '.join(abstracts[a])
```

3.3 Bag of Words Model

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(abstracts).toarray()
X = np.float64(np.array(X))

#normalizes abstracts
for a in range(len(abstracts)):
    X[a,:] = X[a,:]/np.sum(X[a,:])

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans

wcss = []
for i in range(1,81,8):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,81,8),wcss,color = 'black')
plt.title('The Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```

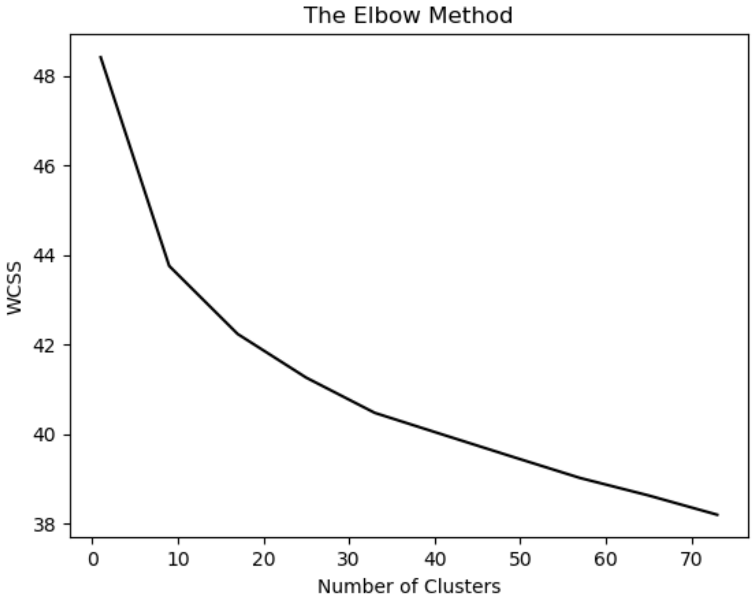


Figure 1: There's not really a distinct "elbow", but 20 clusters seeems reasonable.

3.4 K-means Clustering

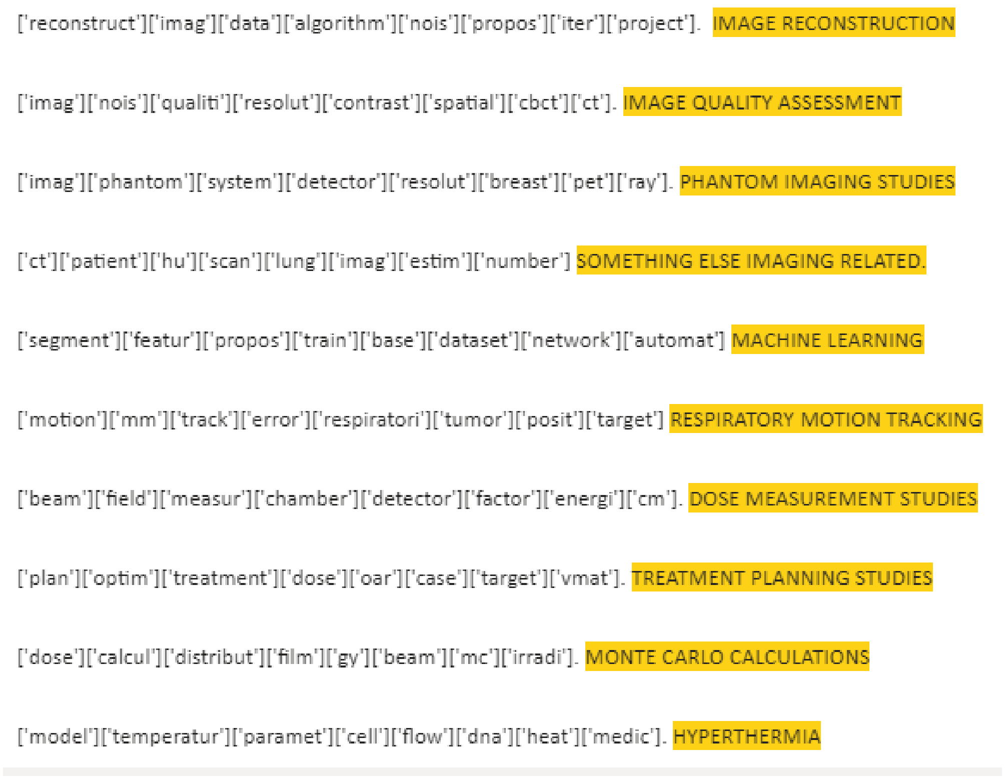
```
# Training the K-Means model on the dataset
clusternum = 20
kmeans = KMeans(n_clusters = clusternum, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
#print(y_kmeans)
```

3.5 Making Some Sense of the Clusters

```
hist, bin_edges = np.histogram(y_kmeans, bins = np.arange(-0.5, clusternum + 0.5, 1))
perc_hist = np.around(hist/paper_number*100,decimals = 2)
decending_list = np.flipud(np.argsort(hist))

#print centers
n_words = 8
perc_sum = 0
centers = kmeans.cluster_centers_
for a in decending_list:
    temp = np.array(centers[a,:]) - np.average(centers,axis = 0)
    stemp = np.flipud(np.argsort(temp))
    perc_sum += perc_hist[a]
    print('\n')
    print('Sum = ' + str(np.around(perc_sum, decimals = 1)) + '%, ' \
        + str(perc_hist[a]) + '% - ', end = '')
    for b in range(n_words):
        ntemp = [0]*len(temp)
        ntemp[stemp[b]] = 1
        print(cv.inverse_transform(ntemp)[0],end = '')
```

And we find clusters such as the following:



['reconstruct']['imag']['data']['algorithm']['nois']['propos']['iter']['project']. IMAGE RECONSTRUCTION

['imag']['nois']['qualiti']['resolut']['contrast']['spatial']['cbct']['ct']. IMAGE QUALITY ASSESSMENT

['imag']['phantom']['system']['detector']['resolut']['breast']['pet']['ray']. PHANTOM IMAGING STUDIES

['ct']['patient']['hu']['scan']['lung']['imag']['estim']['number'] SOMETHING ELSE IMAGING RELATED.

['segment']['featur']['propos']['train']['base']['dataset']['network']['automat'] MACHINE LEARNING

['motion']['mm']['track']['error']['respiratori']['tumor']['posit']['target'] RESPIRATORY MOTION TRACKING

['beam']['field']['measur']['chamber']['detector']['factor']['energi']['cm']. DOSE MEASUREMENT STUDIES

['plan']['optim']['treatment']['dose']['oar']['case']['target']['vmat']. TREATMENT PLANNING STUDIES

['dose']['calcul']['distribut']['film']['gy']['beam']['mc']['irradi']. MONTE CARLO CALCULATIONS

['model']['temperatur']['paramet']['cell']['flow']['dna']['heat']['medic']. HYPERTHERMIA

Figure 2: Word clusters from the dataset show trends in the subfields of medical physics.