HW2_Part2_LDA

October 23, 2022

[1]:
```
from pyspark import SparkConf, SparkContext,SQLContext
from pyspark.sql import SparkSession
from pyspark.ml.feature import Word2Vec,CountVectorizer
from pyspark.ml.clustering import LDA, LDAModel
from pyspark.sql.functions import col, udf, split
from pyspark.sql.types import IntegerType,ArrayType,StringType,StructType,␣
↪StructField
import pylab as pl
```

[2]:
```
def to_word(termIndices):
  words = []
  for termID in termIndices:
    words.append(vocab_broadcast.value[termID])
  return words
```

[3]:
```
#Load your document dataframe here
#================your code here==================
Schema = StructType([
    StructField("twittes", StringType(), True),
])
spark = SparkSession \
    .builder \
    .getOrCreate()
df = spark.read.option("inferSchema", "True").schema(Schema).csv("gs://
↪eecs6893_data/notebooks/jupyter/EECS_6893_Big_Data_Analysis/HW2/stream_data.
↪csv")
spark_df = df.select(split(col("twittes")," ")).
↪withColumnRenamed("split(twittes,  , -1)","words")
#================================================
spark_df.show()
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
22/10/23 15:35:07 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/10/23 15:35:07 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/10/23 15:35:07 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMasterHeartbeat
```

```
22/10/23 15:35:07 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
[Stage 0:>                                                          (0 + 1) / 1]

+--------------------+
|               words|
+--------------------+
|[I, absolutely, A…|
|[Java, Vs, Python…|
|[voulu, un, grec,…|
|[Pareil, Il, pris…|
|[Music, Academy, …|
|[Tarps,, tents,, …|
|[voulu, un, grec,…|
|[We, drive, effic…|
|[Check, out, my, …|
|[Hey,, nice, bone…|
|[lembro, como, so…|
|[WHO, WITH, A, DE…|
|[@Tina69911364, @…|
|[alguem, cria, um…|
|[@Neptvn08, Comme…|
|[une, dinguerie, …|
|[Y, a, une, gross…|
|[Je, te, cache, p…|
|[@JAPANFESS, seta…|
|[Femme, rechercha…|
+--------------------+
only showing top 20 rows
```

[4]:
```
#CountVectorizer
#================your code here==================
# word2Vec = Word2Vec(vectorSize=3, minCount=0, inputCol="words",␣
 ↪outputCol="word2Vec")
# model = word2Vec.fit(spark_df)
# cvResult = model.transform(spark_df)

cv = CountVectorizer(inputCol="words", outputCol="features", minDF=30)
model = cv.fit(spark_df)
cvResult = model.transform(spark_df)
#================================================
```

[5]:
```
#train LDA model, cluster the documents into 10 topics
#================your code here==================
lda = LDA(featuresCol="features",maxIter=1000, k=10, learningDecay=0.45,␣
 ↪optimizer='online', topicDistributionCol='topicDistribution')
ldaModel = lda.fit(cvResult)
#================================================
```

[6]:
```
transformed = ldaModel.transform(cvResult).select("topicDistribution")
#show the weight of every topic Distribution
transformed.show(truncate=False)
```

```
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-------------------------------------------------+
|topicDistribution
|
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-------------------------------------------------+
|[0.009726653482341288,0.003981236916849474,0.00169054864452851,0.01460674894832
4193,0.01900827575610118,0.9355745024303354,0.0016907908124763852,0.010340145272
679279,0.0016905488145869908,0.0016905489217772613] |
|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]
|
|[0.015203679841293566,0.006223050245783997,0.002642487593068984,0.0228321425917
57486,0.8743519256822548,0.0546562467877649,0.002642866124500049,0.0161626252482
5456,0.0026424878588864904,0.002642488026435032]    |
|[0.009726653444138257,0.003981236916849474,0.0016905486445285098,0.580267719638
9255,0.35388298558901055,0.03503882247033361,0.001690790812476385,0.010340144747
373338,0.0016905488145869906,0.001690548921777261]  |
|[0.009726653839090282,0.003981236916849474,0.00169054864452851,0.01460674894838
6645,0.01900918370154153,0.9355735928602485,0.0016907908124763852,0.010340146540
514381,0.0016905488145869908,0.0016905489217772613] |
|[0.015203679841294813,0.006223050245783997,0.002642487593068984,0.0228317315780
68098,0.02971031664624789,0.8992982619416755,0.002642866124500049,0.016162630144
03907,0.0026424878588864904,0.002642488026435032]    |
|[0.015203679841293566,0.006223050245783997,0.002642487593068984,0.0228321425917
57486,0.8743519256822548,0.0546562467877649,0.002642866124500049,0.0161626252482
5456,0.0026424878588864904,0.002642488026435032]    |
|[0.03479856883433165,0.014243475566306889,0.00604819276393379,0.052257847521786
68,0.06800172784236796,0.76951131373266,0.006049059156293889,0.03699342745414232
,0.006048193372343655,0.006048193755833015]         |
|[0.021161700263707495,0.008661740144618037,0.003678026042300924,0.0317790341026
32255,0.04135684264643368,0.8598316134838395,0.0036785529126881072,0.02249643734
5998307,0.0036780264122870366,0.0036780266454945264]|
```

```
|[0.015203679841300862,0.006223050245783999,0.0026424875930689843,0.022831731579
571524,0.029710316666406694,0.6177501728792049,0.0026428661245000494,0.297710719
18484154,0.00264248785886491,0.0026424880264350322]|
|[0.8049014871523601,0.00866174014461804,0.0036780260423009245,0.031779034102912
766,0.04137372338469479,0.07607494585664229,0.003678552912688108,0.0224964373460
01482,0.003678026412287037,0.003678026645494527]     |
|[0.01520367984129463,0.28777218404778854,0.0026424875930689835,0.02283173157802
1982,0.029710316645627097,0.6177491330361196,0.0026428661245000485,0.01616262524
8257546,0.00264248785888649,0.0026424880264350314]   |
|[0.015203679841291905,0.006223050245783999,0.0026424875930689843,0.022831731577
34067,0.02971031663648791,0.05516625457322767,0.0026428661245000494,0.8602946375
229772,0.00264248785886491,0.0026424880264350322]   |
|[0.8049014872931161,0.008661740144618041,0.0036780260423009254,0.03177903410279
996,0.04137372330388928,0.07607494579680606,0.0036785529126881085,0.022496437346
00022,0.003678026412287038,0.0036780266454945277]    |
|[0.011863534664186449,0.004855887918295416,0.0020619508232514114,0.896582221218
845,0.02318981547764406,0.04264863643696211,0.0020622461938703536,0.012611805074
864459,0.002061951030670598,0.00206195116140985]     |
|[0.009726653279675188,0.003981236916849474,0.00169054864452851,0.91521404495429
13,0.01900879314124779,0.03496668976719879,0.0016907908124763852,0.0103401447473
68293,0.0016905488145869908,0.0016905489217772613]   |
|[0.015203682739018218,0.006223050245783999,0.0026424875930689843,0.304378882170
3161,0.5907596744799417,0.05670175551377962,0.0026428661245000494,0.016162625248
269804,0.00264248785886491,0.0026424880264350322]   |
|[0.009726653704901035,0.003981236916849474,0.0016905486445285098,0.915202761871
5419,0.01902007564555555065,0.03496668992041178,0.001690790812476385,0.01034014474
7371533,0.0016905488145869906,0.001690548921777261] |
|[0.034798568834328504,0.014243475566306892,0.006048192763933792,0.0522578475209
9942,0.06800172783180082,0.12687674202443538,0.006049059156293891,0.679627999173
7247,0.006048193372343657,0.006048193755833017]      |
|[0.011863533530844184,0.004855887918295417,0.002061950823251412,0.6394731155837
569,0.2802989217148988,0.04264863696812684,0.0020622461938703536,0.0126118050748
75698,0.0020619510306705982,0.0020619511614098506]   |
+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------------------------------+
only showing top 20 rows
```

[7]:
```
#The higher ll is, the lower lp is, the better model is.
ll = ldaModel.logLikelihood(cvResult)
lp = ldaModel.logPerplexity(cvResult)
print("ll: ", ll)
print("lp: ", lp)
```

```
ll:  -9299.719539892056
lp:  3.308331390925669
```

[8]:
```
# Output topics. Each is a distribution over words (matching word count vectors)
print("Learned topics (as distributions over vocab of " + str(ldaModel.
 ↪vocabSize())+ " words):")
topics = ldaModel.topicsMatrix()
print(topics)
```

```
Learned topics (as distributions over vocab of 40 words):
DenseMatrix([[3.64354276e+01, 1.00004720e-01, 1.00000000e-01, 1.00013217e-01,
              3.27534104e+01, 1.48932363e+02, 1.00000000e-01, 1.00016255e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00000000e-01, 1.00004073e-01, 1.00000000e-01, 1.00000090e-01,
              1.00000213e-01, 1.47440521e+02, 1.00000000e-01, 4.11656658e+01,
              1.00000000e-01, 1.00000000e-01],
             [1.90478278e+01, 1.00006865e-01, 1.00000000e-01, 8.17683588e+01,
              7.85089572e+01, 1.00000050e-01, 1.00000000e-01, 1.00000140e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00000700e-01, 1.00000000e-01, 1.00000000e-01, 1.00000145e-01,
              1.00000460e-01, 9.11496324e+01, 1.00000000e-01, 4.70844861e+01,
              1.00000000e-01, 1.00000000e-01],
             [1.00007944e-01, 1.00017998e-01, 1.00000000e-01, 1.00000000e-01,
              1.00002517e-01, 1.17613057e+02, 1.00000000e-01, 1.00024834e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00000000e-01, 1.00001854e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000001e-01, 1.20602049e+02, 1.00000000e-01, 1.00023122e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00002538e-01, 1.00026772e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000001e-01, 1.13721602e+02, 1.00000000e-01, 1.00010901e-01,
              1.00000000e-01, 1.00000000e-01],
             [4.68392070e+01, 1.00000991e-01, 1.00000000e-01, 4.85941232e+01,
              1.00021885e-01, 1.00000367e-01, 1.00000000e-01, 1.00001561e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00012756e-01, 1.00004636e-01, 1.00000000e-01, 1.00012045e-01,
              8.73581901e+01, 1.00001095e-01, 1.00000000e-01, 1.00003784e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00000000e-01, 1.00007459e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000002e-01, 8.90965351e+01, 1.00000000e-01, 7.16325803e+00,
              1.00000000e-01, 1.00000000e-01],
             [1.00001222e-01, 1.00003759e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000001e-01, 5.09060160e+01, 1.00000000e-01, 3.79432261e+01,
              1.00000000e-01, 1.00000000e-01],
             [1.00005842e-01, 1.00000000e-01, 1.00000000e-01, 3.72079390e+01,
              3.02860361e+01, 1.00000031e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00000000e-01, 1.00021386e-01, 1.00000000e-01, 1.00000000e-01,
              1.00000001e-01, 8.65101798e+01, 1.00000000e-01, 1.00004564e-01,
              1.00000000e-01, 1.00000000e-01],
             [1.00003702e-01, 1.00000000e-01, 1.00000000e-01, 4.61240329e+01,
```

```
 1.58751318e+01, 1.00000081e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00004904e-01, 1.00000000e-01, 1.00000000e-01, 2.12527995e+01,
 3.99037664e+01, 1.00000113e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00005937e-01, 1.00000000e-01, 1.00000000e-01, 1.91583255e+01,
 3.90259688e+01, 1.00000025e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[6.21936203e+01, 1.00000000e-01, 1.00000000e-01, 1.00002907e-01,
 1.00004330e-01, 1.00000736e-01, 1.00000000e-01, 1.00001853e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00004809e-01, 1.00015277e-01, 1.00000000e-01, 1.00046666e-01,
 7.17388283e+00, 4.61789025e+01, 1.00000000e-01, 1.00012223e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00012837e-01, 1.00000000e-01, 1.00000000e-01, 1.55706175e+01,
 4.77389905e+01, 1.00000026e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00002470e-01, 1.00012537e-01, 1.00000000e-01, 1.00000262e-01,
 1.00000001e-01, 4.99528570e+01, 1.00000000e-01, 1.00025779e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00000000e-01, 1.00006089e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 4.32705924e+01, 1.00000000e-01, 1.00007123e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00004646e-01, 1.00005350e-01, 1.00000000e-01, 1.00010712e-01,
 3.75396603e+01, 1.00000020e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00006198e-01, 1.00000000e-01, 1.00000000e-01, 1.00009191e-01,
 4.59439915e+01, 1.00000018e-01, 1.00000000e-01, 1.00000665e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00000000e-01, 1.00003889e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 1.92859706e+01, 1.00000000e-01, 1.87760971e+01,
 1.00000000e-01, 1.00000000e-01],
[1.00000872e-01, 4.23857861e+01, 1.00000000e-01, 1.00000139e-01,
 1.00000197e-01, 1.00001726e-01, 1.00000000e-01, 1.00001564e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00001050e-01, 1.00000000e-01, 1.00000000e-01, 3.67792770e+01,
 1.00006913e-01, 1.00000054e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00000000e-01, 1.00000000e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 4.35374782e+01, 1.00000000e-01, 1.00016655e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00003594e-01, 1.00001335e-01, 1.00000000e-01, 1.00018983e-01,
 3.96110409e+01, 1.00000022e-01, 1.00000000e-01, 1.00006288e-01,
 1.00000000e-01, 1.00000000e-01],
[3.71976551e+01, 1.00001958e-01, 1.00000000e-01, 1.00000775e-01,
 1.00000700e-01, 1.00000126e-01, 1.00000000e-01, 1.00002714e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00001696e-01, 1.00000000e-01, 1.00000000e-01, 2.95344357e+01,
```

```
 1.00012972e-01, 1.00000240e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00000000e-01, 1.00000000e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 3.85722338e+01, 1.00000000e-01, 1.00012319e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00008403e-01, 1.00000000e-01, 1.00000000e-01, 1.00041956e-01,
 3.70084593e+01, 1.00000028e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00000000e-01, 1.00006478e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 1.00011383e-01, 1.00000000e-01, 4.18386447e+01,
 1.00000000e-01, 1.00000000e-01],
[4.14174436e+00, 1.00000000e-01, 1.00000000e-01, 1.93364581e+01,
 1.00009649e-01, 1.41893172e+01, 1.00000000e-01, 1.00001026e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00001159e-01, 1.00000000e-01, 1.00000000e-01, 3.38345259e+01,
 1.00024042e-01, 1.00000017e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00001050e-01, 1.00000000e-01, 1.00000000e-01, 2.96712862e+01,
 1.00022233e-01, 1.00002013e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00005877e-01, 1.00000000e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000001e-01, 3.48458785e+01, 1.00000000e-01, 1.00017094e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00002679e-01, 1.00000000e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000345e-01, 1.00010087e-01, 1.00000000e-01, 3.41650467e+01,
 1.00000000e-01, 1.00000000e-01],
[6.01363575e+00, 1.00000000e-01, 1.00000000e-01, 1.00009851e-01,
 2.46746009e+01, 1.00000025e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01],
[1.00001974e-01, 1.00000000e-01, 1.00000000e-01, 1.40690766e+01,
 1.82952654e+01, 1.00000022e-01, 1.00000000e-01, 1.00000000e-01,
 1.00000000e-01, 1.00000000e-01]])
```