Домашнее задание №13

Асланов А.Б., ИУ9-21М

Есть три текста:
$d_0 : w_0, w_1, w_1$;
$d_1 : w_0, w_1, w_2$;
$d_2 : w_0, w_2, w_2$.
Сделать 5 итераций (проходов по трём текстам) EM-алгоритма

In [21]:
```
import artm

"""
Для работы с BigARTM нужно сначала перевести текст в формат матрицы вхождений слов в документы (bag_of_
words),
а затем его в необходимый для работы библиотеки формат - батчи.
Для того чтобы проделать это, используется конструктор BatchVectorizer, но он принимает на вход только ф
ормат
вида библиотеки Vowpal Wabbit.
Потом предварительно надо привести весь текст в формат Vowpal Wabbit:
|text w0 w1 w1
|text w0 w1 w2
|text w0 w2 w2

Далее полученный конструктор подается на вход алгоритма обучения (метод fit_offline).
fit_offline на каждом проходе обновляет матрицы phi и theta. Используется для маленьких коллекций.

cache_theta = флаг, позволяющий либо запрещающий хранить матрицу theta.
Нужен по причине того, что матрица theta может занимать слишком много места (для больших коллекций),
что часто неприемлемо.

На выходе: матрицы слов в теме и матрица тем в документе.
"""

# подготовка данных
batch_vectorizer = artm.BatchVectorizer(data_path='/Users/user/Desktop/em/doc.txt', data_forma
t='vowpal_wabbit', target_folder='batches')
T = 2 # количество тем
iter_num = 5 # количество итераций
dictionary = batch_vectorizer.dictionary

model = artm.ARTM(num_topics=T, dictionary=dictionary, cache_theta=True)
model.fit_offline(batch_vectorizer=batch_vectorizer, num_collection_passes=iter_num)
phi = model.get_phi()
theta = model.get_theta()
print('После ' + str(iter_num) + ' операций:\n')
print('PHI (матрица слов в темах):\n')
print(phi)
print('\n\n')
print('THETA (матрица тем в документах):\n')
print(theta)
```

```
После 5 операций:

PHI (матрица слов в темах):

                       topic_0   topic_1
(@default_class, w2)  0.000334  0.665316
(@default_class, w1)  0.665499  0.002181
(@default_class, w0)  0.334167  0.332502



THETA (матрица тем в документах):

                 0         1        2
topic_0   0.999991  0.497708  0.00001
topic_1   0.000009  0.502292  0.99999
```

Вывод

После 5 итераций.
Для матрицы слов в теме:
1) К теме t_0 с наибольшей вероятностью относится слово w_1;
2) К теме t_1 с наибольшей вероятностью относится слово w_2.

Для матрицы тем в документе:
1) К теме t_0 с наибольшей вероятностью относится документ d_0;
1) К теме t_1 с наибольшей вероятностью относится документ d_2;