HW4: Topic Modeling

Bhavika Tekwani (btekwani@gmu.edu)

Part 1

I used the *gensim* module in Python to run Latent Dirichlet Allocation for topic modeling.

Parameters used:

```
num_topics = {20, 30, 40}, id2word = dictionary, update_every = 1,
gamma_threshold = 0.001,
iterations = 50, passes=1, chunksize = 10000, minimum_probability = 0.01
```

Here is a tabular representation of the words and topics found depending on the `num_topics` parameter. I present a select number of topics that are interesting. Several top words are common between topics discovered when we change the `num_topics` parameter.

The complete output with 20, 30 and 40 topic outputs is in the attached **nytimes_output.txt** file.

num_topics	**Top words and their weights**	**Topic name**
30	0.022*"company" + 0.017*"million" + 0.015*"zzz_enron" + 0.009*"companies" + 0.007*"firm" + 0.007*"employees" + 0.007*"executive" + 0.006*"executives" + 0.006*"deal" + 0.006*"financial"	Business
	0.019*"computer" + 0.015*"system" + 0.008*"technology" + 0.007*"software" + 0.007*"zzz_microsoft" + 0.006*"program" + 0.005*"company" + 0.005*"user" + 0.005*"data" + 0.005*"information"	Software/ Computers
	0.017*"school" + 0.012*"student" + 0.009*"program" + 0.009*"drug" + 0.008*"patient" + 0.008*"doctor" + 0.007*"women" + 0.006*"health" + 0.006*"percent" + 0.006*"care"	Healthcare
	0.020*"team" + 0.011*"games" + 0.010*"goal" + 0.009*"game" + 0.009*"play" + 0.008*"king" + 0.008*"season" + 0.007*"point" + 0.007*"zzz_olympic" + 0.006*"player"	Sports
20	0.009*"school" + 0.008*"family" + 0.006*"children" + 0.006*"friend" + 0.006*"student" + 0.005*"women" + 0.005*"home" + 0.005*"mother" + 0.005*"book" + 0.005*"son"	Family/Relationships
	0.012*"cup" + 0.010*"minutes" + 0.008*"food" +	Cooking

	0.008*"add" + 0.007*"tablespoon" + 0.006*"oil" + 0.006*"meat" + 0.006*"teaspoon" + 0.006*"pepper" + 0.006*"chicken"	
	0.021*"music" + 0.014*"song" + 0.008*"show" + 0.008*"band" + 0.007*"album" + 0.006*"musical" + 0.005*"play" + 0.005*"record" + 0.005*"sound" + 0.005*"artist"	Music
	0.008*"patient" + 0.007*"drug" + 0.006*"doctor" + 0.006*"cell" + 0.005*"scientist" + 0.005*"disease" + 0.005*"test" + 0.005*"research" + 0.005*"study" + 0.004*"anthrax"	Healthcare
40	0.027*"money" + 0.020*"million" + 0.019*"fund" + 0.014*"percent" + 0.012*"pay" + 0.009*"loan" + 0.007*"bank" + 0.007*"financial" + 0.007*"account" + 0.007*"bond"	Finance
	0.017*"bill" + 0.012*"zzz_congress" + 0.011*"plan" + 0.010*"program" + 0.010*"federal" + 0.009*"government" + 0.008*"billion" + 0.007*"million" + 0.007*"money" + 0.007*"proposal"	Government
	0.012*"cell" + 0.010*"energy" + 0.010*"flight" + 0.008*"gas" + 0.008*"passenger" + 0.007*"oil" + 0.007*"power" + 0.007*"airline" + 0.006*"airport" + 0.006*"pilot"	Travel
	0.007*"art" + 0.006*"show" + 0.006*"book" + 0.006*"zzz_new_york" + 0.005*"artist" + 0.005*"history" + 0.005*"century" + 0.004*"collection" + 0.004*"painting" + 0.003*"zzz_american"	Art & Culture

Part 2

1. Image Data

I used the `LatentDirichletAllocation` class from scikit-learn for the MNIST dataset task. Initially, I added only the test dataset as instructed, but adding the train images definitely helped get better results.

The following parameters were used:

```
n_topics=[10, 20, 50], learning_method='batch', max_iter=[10, 20, 50 ],
n_jobs=3
```

I visualize the top 50 words only but any number of words can be visualized (upto 784).
The images for each iteration and topic are in the **output/mnist** folder.

2. Time Series data

I used gensim's LDA for this task. First, I use the TSMining library in R to convert each time series to a SAX representation.

The parameters used for this are:

```
w = dim(train)[1], a = 7, eps = 0.01, norm = TRUE
```

where w is the number of rows in the dataset.

After obtaining the SAX representation, I have 5 files train1.csv through train5.csv which all contain the SAX representation for each time series in the original dataset (train files only).

After this, we convert the SAX representation to the bag of patterns format. I do this by grouping the SAX representation into "words" of length 5. We can try different lengths of words but for the ease of division into equal sized words, 5 is the best choice based on the length of SAX representations we obtained for each dataset.

Now, we create a corpus for each dataset independently and run LDA. The parameters for LDA are as follows:

```
corpus=bow, id2word=dictionary, num_topics=20, update_every=1,
chunksize=10000, passes=20
```

Here `bow` is the bag of patterns representation of documents (time series), dictionary is a dictionary we created from the set of words ("SAX words") in the train set, `num_topics` has values like 20, 30 and 40.

It is not clear how we can visualize the topics in time series and what to name them.
I have not done the visualization step, but the output for one run of dataset1 is contained in **timeseries_output.txt.**

References

[1] Gensim: Latent Dirichlet Allocation: https://radimrehurek.com/gensim/models/ldamodel.html

[2] Gensim: Corpora and Vector Spaces: https://radimrehurek.com/gensim/tut1.html

[3] Gensim: UciCorpus: https://radimrehurek.com/gensim/corpora/ucicorpus.html

[4] UCI topic modeling on images:
http://psiexp.ss.uci.edu/research/programs_data/exampleimages2.html