# Homework 2

Due: October 8, 2020

You will find in the homework 2 folder a zipped file called bbc-fulltext.zip. Inside the folder are articles from the British Broadcasting Corporation (BBC) collected into 5 major topics. The README file tells you where the dataset comes from.

The homework assignment contains some stuff that you will need to do before answering the questions. For many of the steps, there are functions in both R and Python that will essentially handle them for you. Your job on this part, then, is to identify those functions, and get your data into the right form in order to utilize those functions. In answering the questions, you may be required to re-run the pre-question tasks, so it will be to your benefit to make them easy to modify.

**I will warn you** that there is a lot to do in this assignment. Waiting until the last minute is a really bad idea and, to be honest, will not be met with much sympathy. These are the homeworks that rec letters are written on.

## Pre-question tasks

The basic tasks that you will need to do are:

- Convert the articles into a document-term matrix. This is typically the input required for topic analysis (though not always! So make sure you check the LDA implementation you are using). In this format, your dataset is represented as a matrix with each document being a row of the matrix, and each term being a column. In other words, each row is a bag-of-words "fingerprint" for each document! This is what I call the "high-dimensional" representation.
- Run LDA on your document-term matrix. Both R and Python have functions to do this. But note that there are arguments that you should specify. These include the number of topics.
- Extract the output from the model. You are interested in two things. The first is a collection of probability vectors over words associated with each topic. The second is a collection of probability vectors over topics associated with each document (each row of your matrix). Assuming you don't have as many topics as there are words (which would be a really bad idea), you can see how this second collection is a much lower dimensional representation of each document!

# Questions:

1. The only way to understand topics in a topic model are to look at the words that get a high probability in each topic. Examine your topics in this way and discuss. How many words do you look at in each topic? Hard to say! Look at a variety of lengths. Discuss any concerns you have with having to pick a number of words to look at in each topic. You may want to visualize the top word probabilities in each document with something like a bar chart.
2. Examine and the document topic probabilities. Some things you might want to consider in your exploration: if we consider the topic of a document to be the topic of highest probability, how many documents are about each topic? (and what's more interesting to ask is, does each topic have roughly the same number of documents? Do documents concentrate on certain topics?); what are the top documents in each topic? (i.e. for a given topic, what are the top k documents

in that topic); are there (how many, etc) documents that assign substantial weight to more than one topic?; are there documents that spread weight over multiple documents? (notions like entropy might help you out here. Or you could threshold probabilities and count how many topics in each document exceed that threshold). Whatever investigations you do, I expect this to be comprehensive.

3. When you run a topic model, you have to specify a number of topics. What is the implication of your decision? To investigate this, you should try a range of topic numbers. Some numbers close to the number you originally chose, some others double or triple the number, and some much smaller than the number you chose (of course, you don't have to try every number but pick some close to and some further away (in both directions) from the number you chose)
4. What happens if you keep the number of topics the same but leave out one of the folders in your analysis?
5. When running topic analysis, your topics can change depending on the random initialization that the algorithm starts with. You can control this by setting a seed that is used inside the function you run. How do the topics change? Among the top words in different topics, are there some topics that remain roughly unchanged? Do the distributions of topics over documents change? Or are the documents that, for instance, were strongly about one topic still about one topic (etc). This question will require a thorough investigation.
6. What are your final thoughts on topic modeling?