Modeling text topics with Latent Dirichlet Allocation

In many cases, it is good to think of data as belonging to more than one cluster or category. For example, if we have a model for text data that includes both "Politics" and "World News" categories, then an article about a recent meeting of the United Nations should have membership in both categories rather than being forced into just one.

With this in mind, we will use GraphLab Create tools to fit an LDA model to a corpus of Wikipedia articles and examine the results to analyze the impact of a mixed membership approach.

In this assignment you will

- apply standard preprocessing techniques on Wikipedia text data
- use GraphLab Create to fit a Latent Dirichlet allocation (LDA) model
- explore and interpret the results, including topic keywords and topic assignments for a document

If you are using GraphLab Create

An IPython Notebook has been provided below to you for this assignment. This notebook contains the instructions, quiz questions and partially-completed code for you to use as well as some cells to test your code.

- Download the Wikipedia people dataset in SFrame format:people_wiki.gl.zip
- Download the pretrained models: topic_models.zip
- Download the companion IPython notebook:5_lda_blank.ipynb
- Save all of these files in the same directory (where you are calling IPython notebook from) and unzip the data file and model file.

This assignment will require the use of GraphLab Create, and here is why:

The method used to fit the LDA model is a *randomized algorithm*, which means that it involves steps that are random; in this case, the randomness comes from Gibbs sampling, as discussed in the LDA video lectures. Because of these random steps, the algorithm will be expected to yield slighty different output for different runs on the same data - note that this is different from previously seen algorithms such as k-means or EM, which will always produce the same results given the same input and initialization.

It is important to understand that variation in the results is a fundamental feature of randomized methods. However, in the context of this assignment this variation makes it highly difficult to evaluate the correctness of your analysis, so we will load and analyze a pre-trained model.

You are free to experiment with an LDA implementation of your choice, but for this assessment, you should use GraphLab Create. However, feel free to re-create the analysis done in the IPython notebook for your personal enjoyment.

We recommend that you spend some time exploring your own fitted topic model and compare our analysis of the pre-trained model to the same analysis applied to the model you trained above.

The focus of this assignment is exploration of results, not implementation. We will analyze the fitted model to understand what it has done with our data and whether it will be useful as a document classification system. This can be a challenging task in itself, particularly when the model that we use is complex. We will begin by outlining a sequence of objectives that will help us understand our model in detail. In particular, we will

- get the top words in each topic and use these to identify topic themes
- predict topic distributions for some example documents
- compare the quality of LDA "nearest neighbors" to the NN output from the first assignment
- understand the role of model hyperparameters alpha and gamma

Don't want to install GraphLab Create? Consider using Amazon EC2.