Foundations of Graphical Models: Homework 1

Due: Friday 2019-10-11

The total page limit for problems 1 and 2 is three pages (though you may use extra pages for figures, and your code can be any length.) Please use the LaTeX template on the website. You should zip your writeup and code into one file and submit it on Courseworks.

Problem 1

Implement Gibbs sampling for a mixture model. You can implement a Gaussian mixture model as we discussed in class, or a different mixture model instead.

Apply your code to a real-world data set and discuss what you learned. You can plot and discuss whatever you like. Among the plots, we would like to see $\log p(x_{1:n}, z_{1:n}, \mu_{1:K})$ as a function of iteration. (This is one way to diagnose if the Gibbs sampler converges). As a guiding principle, you may go through one iteration of Box's loop.

You can find examples of data sets on the course website. Feel free to use one of the provided data sets or another data set of your choice.

Getting through this exercise is important for having a good final project and, more generally, becoming fluent in the material. There are many "gotchas" in developing and deploying probabilistic models, which are only learned from experience. (For example, you will want to work in log space, only exponentiating when you need to.)

We expect you to implement your own Gibbs sampler, and to not rely on external implementations such as those offered by TensorFlow or Scikit-learn. Please submit your code in addition to the writeup.

Problem 2

This problem is intended to help you brainstorm ideas for the project. Consider some data. If you have a data set in mind for your final project then we encourage you to use it for this exercise as well.

a) **Variables in the data.** What are the variables in the data and what are some of their relationships to each other? Do you expect some of the variables to be correlated? Do you expect others to be (conditionally) independent?

b) **Latent variables.** What are some latent variables you could introduce to capture the correlations between model variables? What are some latent variables that could summarize aspects of the data? What other latent variables could be hidden in the data?

c) **Research question.** Formulate several questions you might be able to answer with the data. Write down the three most interesting ones.

Online Data Sets

Senate One of the online data sets contains Senate voting data. After unzipping, you'll find two files in the folder `senate`: `votes.csv` and `senators.txt`.

Each of the $n = 103$ rows of `votes.csv` contains the voting record of a different member of the 113'th session of the United States senate. The columns correspond to $d = 657$ bills that were voted on: 1 indicates a 'yea' vote, 0 indicates a 'nay' vote, and -1 indicates that the particular member did not vote.

`senators.txt` contains the names of the 103 senators, in the same order they appear on `votes.csv`. Each senator's name contains his/her political party (D, R, or I) and state; for example, Schumer (D-NY) indicates that Schumer is a Democratic senator from New York.

Given the binary nature of votes, it may make sense to use a Bernoulli mixture model with Beta priors on the means (after appropriately accounting for absent voters).

AP The other provided data set contains the text of 2,246 articles from the Associated Press. After unzipping, you'll find three files in the folder `ap`: `ap.dat`, `ap.txt`, and `vocab.txt`.

`ap.txt` is an XML file that contains the full text of every article.

You'll probably want to work with `ap.dat`, which contains the counts of each word for each article. Every line is a different article in a bag-of-words format. The first number in each line is the total number of words in that article. Following this number, the rest of the line contains word counts in the format `word_index:count`. For example, the line "`5 0:1 5:2 140:1 2031:1`" indicates a 5-word article that has 1 occurrence of word 0, 2 occurrences of word 5, 1 occurrences of word 140, and 1 occurrence of word 2031.

Finally, the file `vocab.txt` contains a list of each word, zero-indexed to match the indices in `ap.dat`.

Since each article is a vector of counts, it may make sense to use a Poisson mixture model with Gamma priors on the means. Note that you may want to remove the most common words (stopwords) along with very rare words in order to ease computation and produce more interpretable mixtures.