Nathan Watts

Gaussian LDA report

My goal from the outset of this project was to compare the results of Vanilla LDA with a variant called “Gaussian LDA.” The basic idea of this variant is to replace the one-hot vector encoding words with a word embedding generated by Word2Vec or GloVe (I used the former).

The sampler was not as difficult to implement as expected in and of itself-- it was fairly trivial to remove a single word before calculating the distribution over topics, than sample randomly to add it back in. What was difficult was calculating this multivariate student-t distribution. It caused problems with overflows and floating-point precision. I ended up having to work almost entirely in log space, then exponentiate.

I also had problems with the running time of the algorithm. I tried to alleviate this by storing as much as possible-- inverse covariances, log determinants, squared sums, etc. Anything which would take longer to compute as needed for the t-distribution than to retrieve from a matrix. This helped, marginally, but the running time of the algorithm on the full dataset still remained approximately 50 hours. Even after dramatically reducing the dataset size it was still several hours. In the end the best approach seemed to be to save intermediate checkpoints for verification so I could iterate on a more reasonable timetable.

I was unable to duplicate the results of the paper. When run, every word is sorted into the same topic. This is because the multivariate-t distribution favors topics with a larger number of words allocated to them, and thus the topic which is randomly assigned the most words during initialization will only grow. This also occurs with the author’s implementation when tested, in spite of the results shown in their paper. Were I to repeat this project, I would likely begin by attempting to replicate their results exactly using their own implementation to verify that the results are legitimate-- (although perhaps this is simply a result of the embeddings I chose.)