Probabilistic Models and Machine Learning: HW 2 Report
Yusuf Efe (me2799)
October 30, 2022

Problem 1)

For this homework, I firstly implemented Gibbs Sampling from scratch and it was quite helpful to understand the theoretical derivation steps and understand the underlying principles. After managing to pass all unit tests, I started working on data preprocessing. I chose CIFAR-10 data set to work on a mixture model, since all the data samples have just one label (not more than one like topic modelling examples). To make the data easy to work on, I applied PCA and reduced its dimensions while keeping the most of its variance. After that, I used this data set and fit a neural network on it to verify the success of the data preprocessing (to see how much loss it leads to because of the dimensionality reduction). After the data preprocessing, the model still achieved around 70 percent, which was enough for a 10-class data set. Following this, I started trying my model on this data set. I did know that the convergence of Gibbs sampling was going to be an issue, that is why I tried to keep some priors fixed, and reduced the data dimensions as much as possible. Since I knew that there are equal number of samples for each one of 10 classes, I kept all θ values equal and constant, 1/10. Thus, these parameters were not being updated during the training, which was reducing the complexity of the algorithm. Also, I reduced the number of features for each sample from 3072 to just 5, still keeping the at least 50 percent of the data variance.

Although I did all the unit tests and sanity checks, the loss kept increasing during the training. Actually, considering the difference between VI methods and Gibbs sampling, I was expecting Gibbs Sampling to converge later than VI, which is a pure optimization algorithm. However, I could not deal with the slowness of the convergence and the model never has led to the desirable results for the Gibbs Sampling Method and the PA-applied CIFAR-10 data set. As a result, I decided to use some libraries for LDA to work on topic modeling news headings data.

Topic modeling with LDA on news headings:

There were mainly three experimentation I did using the LDA model.

1. I checked the perplexity (the average negative log likelihood) as a function of the iterations to evaluate the converging of the model. Since calculating the perplexity takes time as much as one-epoch-long-training, I calculated the perplexities once in each two epochs. I collected 5 data points for the epochs 2-4-6-8-10.

2. I kept the 20 percent of the data as the held-out set and checked the perplexity of it, too. As expected, these loss values were much higher than the training set's values

3. Thirdly, I trained the model for two different setting for 10 epochs each. In the first, I made the topic number 4, in the second setting I did 8 topics. So that I could compare topic successes. At each case, I printed the most occurring 7 words of each topic.

1) Checking the Perplexity over iterations:

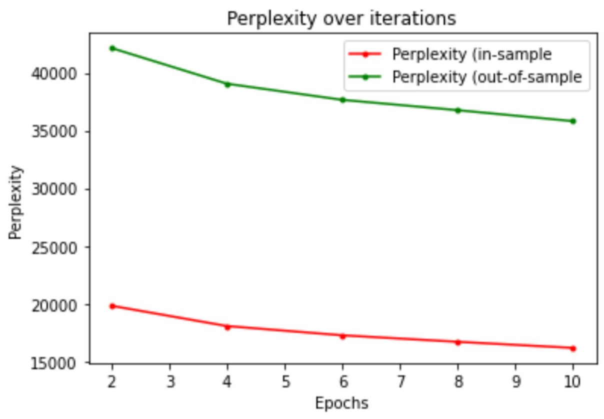

Figure 1: Perplexity on in-sample and out-of-sample data over iterations

Here, we can see that the algorithm converges easily (since it is an optimization algorithm rather than sampling.) Also, we notice a huge difference between the perplexity of in-sample and out-of-sample portions. Still, we see a similar form in both of them. Here, it is also good to see the topic distributions as a function of the epochs, which is provided below:

After 2nd epoch:

Topic 0: ['new', 'power', 'water', 'day', 'closer', 'council', 'final', 'win', 'rain', 'wins']
Topic 1: ['interview', 'north', 'south', 'new', 'police', 'west', 'australia', 'council', 'aid', 'cup']
Topic 2: ['man', 'police', 'crash', 'charged', 'car', 'killed', 'missing', 'court', 'murder', 'woman']
Topic 3: ['govt', 'abc', 'market', 'new', 'rural', 'plan', 'council', 'report', 'rise', 'business']

After 6th epoch:

Topic 0: ['interview', 'day', 'australia', 'cup', 'win', 'new', 'world', 'says', 'final', 'drug']
Topic 1: ['man', 'police', 'crash', 'killed', 'court', 'charged', 'death', 'woman', 'missing', 'car']
Topic 2: ['govt', 'abc', 'water', 'new', 'rural', 'farmers', 'council', 'plan', 'police', 'urged']
Topic 3: ['new', 'council', 'police', 'plan', 'centre', 'blaze', 'fears', 'water', 'interview', 'home']

After 10th epoch:

Topic 0: ['police', 'man', 'crash', 'charged', 'car', 'killed', 'missing', 'death', 'woman', 'dies']
Topic 1: ['health', 'govt', 'new', 'rural', 'child', 'budget', 'says', 'council', 'news', 'funding']
Topic 2: ['water', 'council', 'new', 'govt', 'plan', 'abc', 'weather', 'says', 'pm', 'report']
Topic 3: ['interview', 'win', 'cup', 'world', 'court', 'test', 'final', 'guilty', 'murder', 'trial']

As for the comparison, here we can notice that some mostly occurring topics share multiple topics. For example, topic 0 and 3 are similar in the epoch-2 case. However, as we increase the number of the epochs, now the words in the topics are closer to each other and each topic is almost disjoint from the other topics.

The last comparison is that, I run the algorithm for 10 epochs for 2 different scenarios: 4 topics and 8 topics cases:

After 10th epoch (4 topics):

$$\begin{cases}
\text{Topic 0: [’police’, ’man’, ’crash’, ’charged’, ’car’, ’killed’, ’missing’, ’death’, ’woman’, ’dies’]} \\
\text{Topic 1: [’health’, ’govt’, ’new’, ’rural’, ’child’, ’budget’, ’says’, ’council’, ’news’, ’funding’]} \\
\text{Topic 2: [’water’, ’council’, ’new’, ’govt’, ’plan’, ’abc’, ’weather’, ’says’, ’pm’, ’report’]} \\
\text{Topic 3: [’interview’, ’win’, ’cup’, ’world’, ’court’, ’test’, ’final’, ’guilty’, ’murder’, ’trial’]}
\end{cases}$$

After 10th epoch (8 topics):

$$\begin{cases}
\text{Topic 0: [’interview’, ’win’, ’cup’, ’final’, ’clash’, ’tigers’, ’open’, ’england’, ’play’, ’title’]} \\
\text{Topic 1: [’man’, ’police’, ’court’, ’charged’, ’murder’, ’death’, ’missing’, ’accused’, ’charges’, ’woman’]} \\
\text{Topic 2: [’iraq’, ’killed’, ’news’, ’day’, ’pakistan’, ’kills’, ’dead’, ’troops’, ’blast’, ’abc’]} \\
\text{Topic 3: [’council’, ’govt’, ’new’, ’plan’, ’abc’, ’weather’, ’union’, ’labor’, ’says’, ’workers’]} \\
\text{Topic 4: [’market’, ’prices’, ’live’, ’rain’, ’east’, ’share’, ’markets’, ’price’, ’australian’, ’firefighters’]} \\
\text{Topic 5: [’interview’, ’world’, ’asylum’, ’new’, ’cup’, ’wins’, ’australia’, ’smith’, ’seekers’, ’league’]} \\
\text{Topic 6: [’water’, ’health’, ’govt’, ’new’, ’plan’, ’funding’, ’boost’, ’urged’, ’council’, ’qld’]} \\
\text{Topic 7: [’crash’, ’car’, ’dies’, ’man’, ’fatal’, ’accident’, ’police’, ’driver’, ’plane’, ’killed’]}
\end{cases}$$

Here, we notice that as we increase the number of the topics, there some topics that are kind of duplicates of others. That is why, making a check of the diversity of the topics is an important step to make sure to have correct number of topics. One can iteratively check the similarity score of topics to decide on the correct number of topics, with a binary search methodology.