Movie Review Sentiment Analysis Project

ECE 684: Natural Language Processing, Fall 2020

Guillem Amat, Sebastián Soriano Pérez

1 Introduction

In this report we explain how we created a synthetic dataset using a generative probabilistic language model, and then used two different models to perform *Sentiment Analysis* on both this synthetic dataset and a public one containing movie reviews.

1.1 Sentiment Analysis

Sentiment Analysis, also called opinion mining, refers to the application of Natural Language Processing and AI techniques to extract emotion scores from subjective pieces of text data. It is commonly used in business settings to understand customer segments, measure brand image or detect sentiment in the news feed or social media.

There are a range of different algorithms used to predict sentiment. Some of the most famous ones include Naïve Bayes, Multi-Class Logistic Regression or Support Vector Machines. Lately, the irruption of deep learning models has taken center stage, and different architectures of Recurrent Neural Networks and Transformers have achieved State-of-the-Art results.

2 Dataset

2.1 Real dataset

The *IMDB Movie Review Collection* is a publicly available dataset used for binary sentiment classification that contains considerably more data than previously available ones. It is a simplified version of Stanford's AI Lab dataset of movie reviews from IMDB, this one being 20% of the original one's size, which contained 50.000 movie reviews, evenly split between positive and negative. It provides the data under bag of words and raw text formats and it contains both labeled and unlabeled examples. The dataset can be found in the following link: https://ai.stanford.edu/~amaas/data/sentiment/

2.2 Synthetic dataset

In order to generate an additional synthetic dataset of movie reviews we used a generative Latent Dirichlet Allocation model that took the IMDB Movie Review dataset as an input. We split the IMDB Movie Review dataset into a dataset with only positive reviews and another one with only negative reviews. Each of these subsets was later fitted to a parameter-inference LDA model in order to model four topics contained in it. These inference-LDA models were used to learn the most relevant vocabulary and the topic-word matrices for the positive and negative

datasets, as well as the topic distribution in each of them. Finally, the mean length of both the positive and negative reviews was calculated.

With these parameters we were able to create simple synthetic positive and negative reviews datasets separately, each with 5000 documents or reviews. We merged and shuffled both datasets along with their corresponding labels (0 if they were created using the negative review dataset, 1 if they used the positive one). Finally, we split the 10000 labeled dataset into a synthetic training dataset and a synthetic test dataset, with 5000 documents or reviews each. We used these synthetic datasets to train an LDA-based classifier as well as an LSTM neural network to compare their performance to the original IMDB Movie Review dataset.

3 Models

3.1 Latent Dirichlet Allocation (LDA) Models

The LDA models require a collection of documents (our datasets of movie reviews in this case) from which it is possible to model a set of topics that explain why some of the documents are similar to each other. LDA models describe each document as a mixture of a defined number of topics, where each topic is defined by a specific distribution of words in it.

3.1.0 Generative Probabilistic LDA Model

We processed the IMDB Movie Review training set in order to generate synthetic data. The generative probabilistic LDA model requires the following input parameters to generate synthetic data:

- A vocabulary of words or tokens from which to draw in order to build new documents or reviews
- A topic-word matrix that indicates the distribution of words (columns) in each topic (rows) in the collection of documents or corpus
- A topic distribution vector that indicates the distribution of topics in the corpus
- A mean length (number of words or tokens) for the documents or reviews to be generated

As previously described, in order to create both positive and negative commentaries, an assumption was made that these two classes differed enough for a parameter-inference LDA model to model distinctly different topics when fitted for four topics on each subset. The output of this first preprocessing step was four distinct topics for positive reviews, with the distribution of the main words found in each, and four distinct topics for negative reviews, with the distribution of the main words found in each. A vocabulary for positive reviews was built from the main words found in the four inferred topics, and a vocabulary for negative reviews was built similarly. The topic-word matrices and topic distribution vectors could be inferred from this point, and the mean length of the reviews was easily computed for both positive and negative reviews.

Two separate generative probabilistic LDA models were run, one with the positive reviews parameters and one with the negative reviews parameters. In this way, we were able to create 5000 positive reviews or documents and 5000 negative reviews or documents. We then merged and shuffled the synthetic positive and negative reviews and split them into a training and a test synthetic dataset. Since the vocabulary used to generate the synthetic reviews was limited to the main words found in each topic, the language used in them is composed of random words without appropriate syntax. The LDA generative model does not take word order into account and thus the resulting synthetic datasets are very simple. A couple sample reviews that illustrate this approach include:

- Positive review: *"much man rob good well best time best human much much well good great great good great great time get [...]"*
- Negative review: *"make people really much much real time time bad time even make man good plot bad really first good time [...]"*

3.1.1 Parameter-Inference LDA Model and Logistic Regression Classifier

We later fitted two parameter-inference LDA models to infer four main topics in each of our datasets, the real reviews dataset and the synthetic reviews dataset. The parameter-inference LDA model returns the distribution of the most relevant words in each of the four modeled topics after fitting the data, and is capable of predicting the topic distribution for any additional or unseen document or review. We fitted the first model using the real reviews training dataset and used the trained model to obtain the topic distribution of each one of the reviews in both the training and the test datasets. With these four topic distributions, we created a new dataset with four variables, in which each one of the variables represents the weight one of the modeled topics has on each one of the reviews. We followed the same steps to fit a model to the synthetic data and transform both the training and test synthetic datasets into a four-variable dataset where each variable represents the weight of one review for each modeled topic.

Having each review now represented as a distribution of weights across four different topics, and using the corresponding labels in the real and synthetic datasets, it is now possible to train a simple classifier model to try to predict whether a review is positive or negative according to its topic distribution. The classifier we used was a logistic regression model with one output node where 0 represented a negative review and 1 a positive review. Two classifiers were trained, one using the real training dataset and another using the synthetic training dataset. The predictions, accuracy, and loss were obtained for both the training and test splits of each, the real and the synthetic datasets. The ROC curves for models trained on the real and synthetic datasets calculated on both the training and test splits can be observed in *Figure 3.1.1.1*.

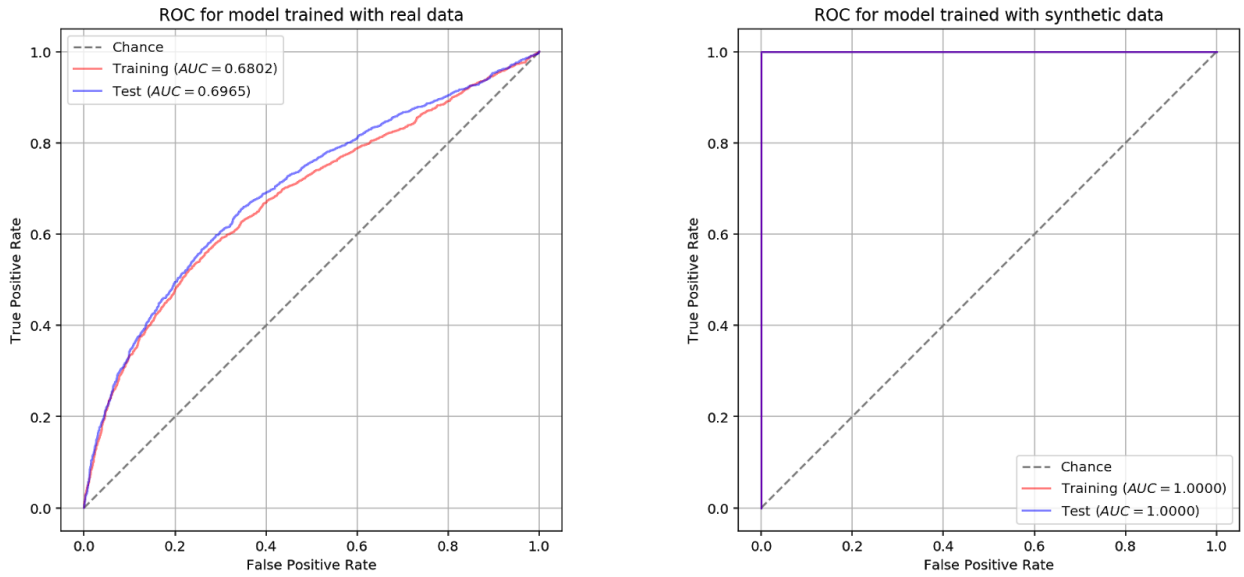

Figure 3.1.1.1: Left: ROC plot for the prediction scores on the training and test splits of the real reviews dataset. Right: ROC plot for the prediction scores on the training and test splits of the synthetic reviews dataset.

3.2 Discriminative Neural Network: Long-Short Term Memory

One of the main problems surrounding plain-vanilla Neural Networks is that long matrix products tend to result in vanishing and exploding gradients. The presence of important elements at the beginning of a sequence, the fact that some inputs to the network do not carry any information or the possibility that there is a logic breakdown in a sequence, are elements that a Recurrent Neural Network is not designed to handle well.

The Long-Short Term Memory and its sibling the Gated Recurrent Unit are special types of RNN that support gating hidden states, which help mitigate the aforementioned problems.

An LSTM can be seen as an enhanced version of a GRU. LSTMs are composed of a memory cell that is governed by a system of gates: The output gate, the forget gate and the input gate. The output gate reads the entries from a cell. The input gate discerns whether to read data into the cell or not. The forget gate resets the content of a cell. This system of gates allows the LSTM to choose whether to remember or forget data. In a similar fashion to a GRU, the output of these gates is combined to create a Hidden State and an Output.

Our specific LSTM architecture processes batches of reviews of a specific length. We first have an embedding layer that adds a third dimension to every word in every review. The result of the embedding layer is combined with the hidden layer from the previous iteration and fed to a 2-layer deep LSTM cell, which generates a Hidden State and an Output. The Output is flattened and passed to a Linear layer. Finally a Sigmoid function is applied to the Output, which generates a vector of predictions. We also added a dropout layer with a value of 0.2 to prevent overfitting and we clipped the gradients to avoid exploding ones.

4 Results

The table below summarizes our results for the synthetic data generated with the probabilistic LDA and the IMDB Movie Review Dataset. The values correspond to the results from our deterministic LDA and LSTM:

Dataset	Performance	LDA	LSTM
LDA Synthetic Data	Train Loss	0.0032	0.004
	Test Loss	0.0032	0.003
	Train Accuracy	100.0%	100.0%
	Test Accuracy	100.0%	100.0%
IMDB Movie Reviews Dataset	Train Loss	0.6360	0.17
	Test Loss	0.6327	0.63
	Train Accuracy	64.4%	91.2%
	Test Accuracy	64.9%	80.4%

5 Conclusions

LDA and LSTM are two very good model choices for sentiment analysis.

LDA can be used both as a generative and a deterministic model:

- As a generative model our approach was to use a simplified vocabulary to reduce time and computational complexities. In consequence, our synthetic data was too simple and did not follow appropriate syntax or grammar. Several improvements can be made to the generative LDA model in order for it to include a more comprehensive vocabulary, and a N-gram-term matrix alongside a deterministic reordering algorithm could be implemented to rearrange the synthetic reviews so they would look more natural.
- The deterministic model could be improved by incrementing the number of topics to be modeled, so that a larger amount of information could be passed onto the classifier in order to obtain more accurate predictions. On the other hand, the perfect accuracy obtained with the synthetic data reflects the fact that it was generated using a simplified and distinct vocabulary for each, the positive reviews and the negative reviews, separately. As the generative model improves, the classifier will have a more difficult task, and increasing the number of topics to model with the parameter-inference LDA model will also improve the classifier's predictions.

An LSTM, our deterministic neural network choice, has many advantages over LDA:

- Our LSTM offered superior predictive power across the board. It was able to detect the sentiment of reviews that were not clearly positive or negative. Given enough computational power and time, we are confident that we would have been able to achieve accuracy values close to or above 90% in the test set.
- LSTMs have the potential to be scaled into more complex neural network architectures. We could perhaps just allow for more cell layers, or we could instead chain the output of the LSTM cell to more linear layers. We could even use an LSTM cell as a part of an Encoder-Decoder model, which have shown great results for Sentiment Analysis.

Nevertheless, there are many drawbacks to using an LSTM for Sentiment Analysis too:

- The time required to get the model running so it achieves decent results is orders of magnitude greater than the time required to train LDA.
- The model suffers from a severe lack of interpretability. Although we know how an LSTM cell works, how it learns and updates weights and the meaning of its different components, it is hard to decipher how it makes decisions based on its weights.
- The LSTM is unstable in the training stage and can easily fall into local minima. Training it is an art that requires both skill and luck.
- LSTMs can easily overfit to the data. Our LSTM started overfitting at epoch 5, although there are many mechanisms like weight decay or a stronger dropout value that we could have applied to prevent that.

6 References

Ye, Jingyi & Jing, Xiaojun & Li, Jia. (2018). Sentiment Analysis Using Modified LDA. 10.1007/978-981-10-7521-6_25.

Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher (June 2011). Learning Word Vectors for Sentiment Analysis.