Dimensionality Reduction for Text Classification with Latent Dirichlet Allocation*

Kunyu He†
Harris School of Public Policy
The University of Chicago
Chicago, IL 60637
kunyuhe@uchicago.edu

Abstract

The curse of dimensionality is not rare for text classification tasks, and overfitting can be a direct result. This project attempts to incorporate a topic modeling technique, Latent Dirichlet Allocation (LDA), as a preprocessing step in the classification pipeline to address the issue. Using extracted topic distributions as encoding vectors, each document is represented as a linear combination of the latent topics. Empirical results show that although incorporating LDA might harms model performance, it can help reduce training time and address overfitting significantly.

1 Introduction

In the text classification problem, we wish to classify a *document* (a sequence of n_i words denoted by $w = (w_1, w_2, ..., w_{n_i})$) into two or more mutually exclusive classes.Use cases include labeling user reviews according to their sentiment, categorizing unstructured documents into standardized subcategories. Since text data is typically not in a format that machine learning models can handle, we need to convert it into its numeric representation. *Bag-of-words* comes in handy. It discards most of the structure of the input text and forms the *Document-term matrix (DTM)*, with documents on the row, unique words on the column, and each entry represents the frequency of each *term* in each document in the *corpus* (a collection of M documents denoted by $D = \{\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_M}\}$).

Treating individual words as features yields a informative but very large feature set (Joachims, 1999). Large vocabulary size can create problems like data sparsity, which is problematic for out-of-sample tests: for documents with words that did not appear in the training corpus, maximum likelihood estimates of the multinomial parameters would assign zero probability to them. On the other hand, when the DTM is high-dimensional ($M < \bar{N}$, where $\bar{N}$ is the vocabulary size), classifiers might suffer from the curse of dimensionality, *overfitting*. Adding to that, sparse matrices can make subsequent computations harder and compromise the efficiency of data storage.

Significant progress has been made on this problem and researchers have proposed many approaches to find lower-dimensional representations of the DTM. *Latent semantic indexing (LSI)* (Deerwester et al., 1990) is one of them. LSI applies *singular value decomposition* to identify a subspace approximation of the *term frequency-inverse document frequency (tf-idf)* (Salton and McGill, 1983) rescaled DTM. It also captures some aspects of basic linguistic notions through the linear combinations of the original *tf-idf* features. `From a generative probabilistic perspective`, one significant step forward is the *probabilistic LSI (pLSI)* model (Hofmann, 1999). However, for pLSI we need a distribution for

*Project repository: https://github.com/KunyuHe/Dimensionality-Reduction-for-Text-Classification-with-LDA

†Kunyu holds a Bachelor of Science from Nanjing University and is pursuing his Master of Science in Computational Analysis and Public Policy. LinkedIn profile: https://www.linkedin.com/in/kunyuhe/, GitHub profile: https://github.com/KunyuHe.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

each document, resulting in a model where the set of parameters grow with the size of the corpus and we cannot associate topics to new documents outside the training corpus. Blei et al. proposed *Latent Dirichlet allocation (LDA)* model as an improvement (Blei et al., 2003).

The report is organized as follows. In Section 2 we describe LDA. We demonstrate the workflow and summarize the empirical results in Section 3 and discuss the caveats of the project in Section 4.

2 Latent Dirichlet Allocation

The idea of LDA is that documents are represented as random mixtures over latent topics *(latent multinomial variables)*, where each topic is characterized by a distribution over words. LDA assumes the following generative process for each document w in a corpus D (Blei et al., 2003):

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim Multinomial(\theta)$.
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

The key inferential problem to be solved is computing the posterior distribution of the hidden variables given a document:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}$$

The posterior distribution is intractable for exact inference (Dickey, 1983). However, a wide variety of approximate inference algorithms can be considered, including *Laplace approximation*, *variational approximation*, and *Markov chain Monte Carlo* (Jordan, 1999). Those are beyond the scope here.

Given the hyperparameters K (the assumed number of latent topics, $K \ll \bar{N}$), α and β, we can obtain an approximated posterior for each document and form the *document-topics matrix (DtM)* of shape $M \times K$. DtM is a lower-dimensional representation of the DTM, and it can be used as input for document classifiers like *Logistic Regression*, *Support Vector Machine*, or *Naïve Bayes*.

3 Problem Definition and Empirical Results

This project incorporates LDA as the dimensionality reduction step in text classification pipelines. It also compares the empirical result pipelines that only apply *tf-idf* rescaling. The workflow for using LDA or LSI for dimensionality reduction is described in Figure 1 below.

First, we apply the same cleaning process on both training and test corpus and transform them into their numeric representations. The cleaning process incorporates bad-of-words model with *bigrams* (pair of words), and it also removes the punctuation, lemmatizes the words, and excludes a list of standard and extended stopwords. Second, we split the original training set into training and validation set. We then fit a set of LDA (or LSI) models on the training DTM (or *tf-idf* rescaled training DTM) with different K, transform it into its lower-dimensional representation DtM, and use it as input for a linear classifier. We set the document-topic prior α to $K/50$ and the topic-word prior β to 0.01 according to empirical research (Wei and Croft, 2006). With grid-search on the validation set, we find the best K for the LDA model and the best set of hyperparameters for the classifier through maximizing the evaluation metrics. Third, we refit the pipeline on *the original* training DTM and make predictions with the transformed test DTM.

We use a dataset of 50,000 movie reviews from the IMDb (Internet Movie Datbase) website collected by Andrew Maas (Maas et al., 2011). Each review in the dataset is labeled "positive" or "negative" according to its rating of the movie from 1 to 10. Reviews with a score of 6 or higher are labeled as positive. Half of the dataset is used as training set and the other half as test set. Both are perfectly *balanced*. Our goal is to build a text classification pipeline that distinguish positive reviews from

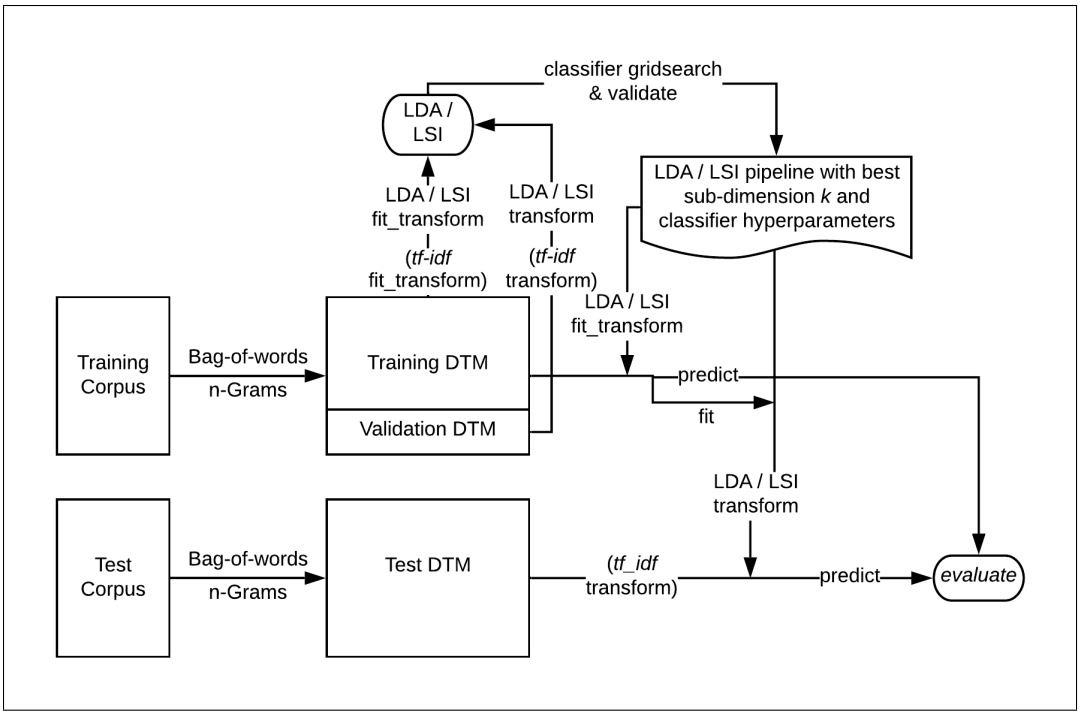

Figure 1: Text Classification Pipeline using LDA or LSI as a Preprocessing Step

negative ones. The classifier should generalize well enough on unseen movie reviews and tag them with correct sentiment labels.

We use *Logistic Regression* as our classifier. It is easy to interpret, fast to train, and perform well on large and high-dimensional datasets. The evaluation metrics we use is *AUC*, as the *false positive rate* against the *true positive rate* (*recall*). K, α, regularization parameter λ and other hyperparameters are determined by optimizing AUC on the validation set. We then refit the pipeline on the original training set, transform the test set, and report the training and test performance.

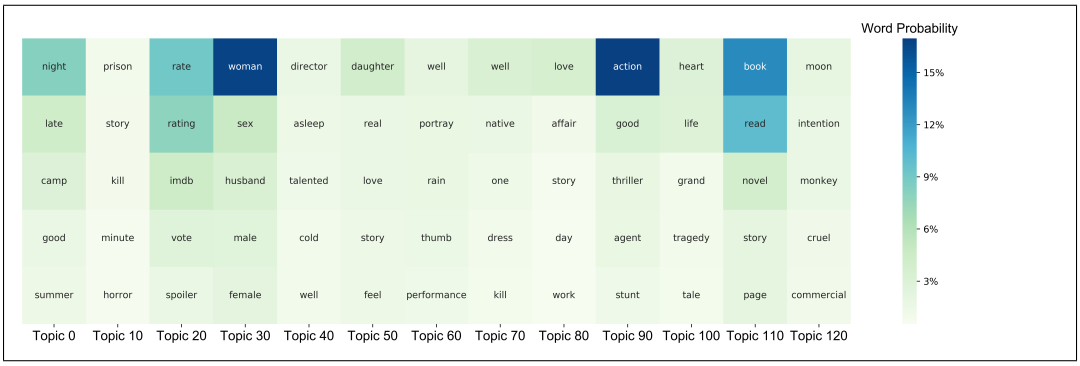

Figure 2: Illustration of the Output of a LDA model

Figure 2 illustrates the output of a LDA model. It shows the top 5 words (sorted by conditional probability on the corresponding topic) in each of the 13 casually chosen topics, out of the 130 topics extracted through the LDA step of the "best" classification pipeline. From these words, we imply that the latent topics are extracted based mostly on movie genres. LDA takes an unsupervised approach, and the latent semantic structure that DtM represents would not necessarily lead us to better classification results. Figure 3 shows the *precision-recall trade-off* and the *Receiver Operating Characteristic (ROC)* curve of the best pipeline trained with the whole training corpus.

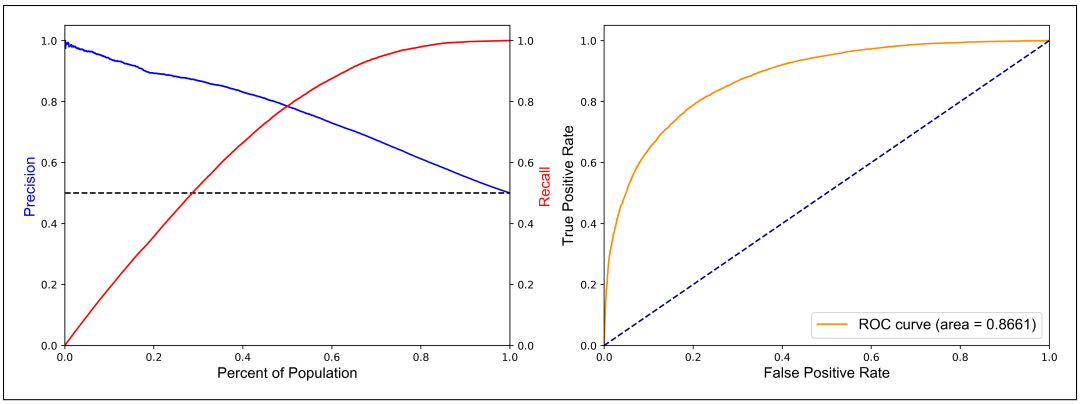

Figure 3: Precision-recall Trade-off and ROC curve for the "Best" Classification Pipeline

As an experiment, we also use different proportions of the original training corpus to train the classification pipeline that incorporates LDA. We also list the results from pipelines that only use *tf-idf*. Figure 4 compares how their test AUC, difference between test and training AUC, number of features, and training time change with increased proportions of the original training corpus used.

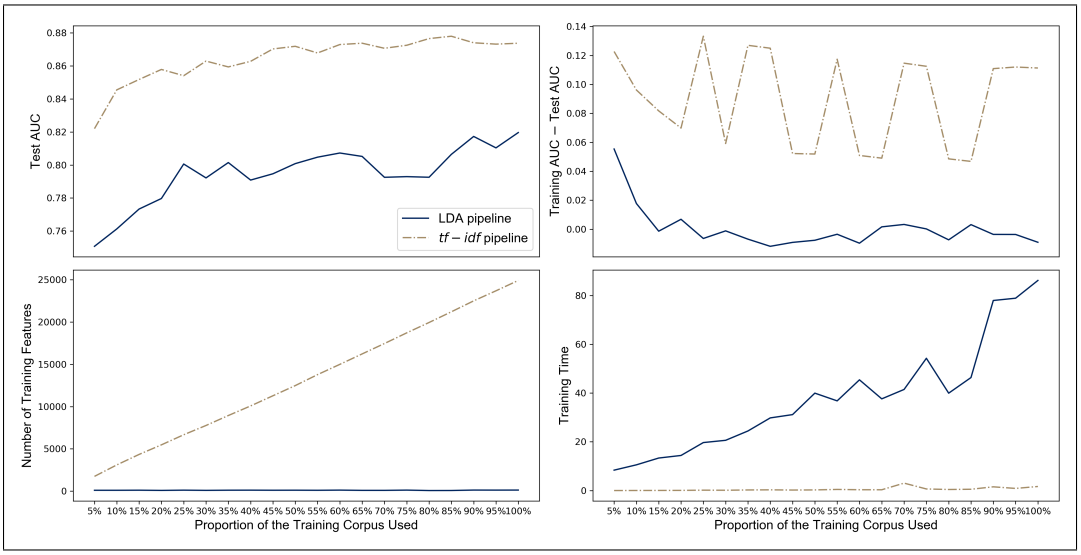

Figure 4: Comparing LDA Pipeline with tf-idf Pipeline on Different Training Corpus Size

Apparently, test AUC for the classification pipeline with LDA preprocessing is lower by approximately 0.07. Besides, its training time is linear to the training set size, while that of the *tf-idf* remains nearly constant. On the bright side, difference in test and training AUC is smaller for the LDA classification pipeline by 0.06 and is robustly decreasing with sample size. In addition, the reduction in dimensionality is very significant.

4 Discussion

In this project we incorporate LDA as a dimensionality reduction tool in the text classification pipeline. We introduce the LDA model, describe the workflow of the classification pipeline with LDA/LSI preprocessing, and evaluate the pipeline through a sentiment classification task on the IMDb movie review data. From empirical results, incorporating LDA can harm model performance and lead to longer training time. However, it reduces the dimensionality of training data significantly and can help reduce overfitting.

The project has several caveats. Due to the long training time, we did not use grid search to find the best decision threshold for the classifier and might failed to get the optimal AUC. Since grid search for LSI takes really long, we did not include LSI into the classification pipeline performance comparison.

References

[1] Blei, D. M. & Ng, A. Y. & Jordan, M. I. (2003) Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**:993–1022.

[2] Dickey, J. (1983) Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association* **82**:773–781.

[3] Hofmann, T. (1999) Probabilistic latent semantic indexing. *Proceedings of the Twenty-Second Annual International SIGIR Conference*.

[4] Jordan, M. (1999) *Learning in Graphical Models*. Cambridge, MA: MIT Press.

[5] Joachims, T. (1999) Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA: MIT Press.

[6] Maas, A. L. & and Daly, R. E. & Pham, P. T. & Huang, D & Ng, A. Y. & Potts, C. (2011) Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

[7] Salton, G. & McGill, M. (1983) In *Introduction to Modern Information Retrieval*. Chicago, IL: McGraw-Hill.

[8] Wei, X. & Croft, W. B. (2006) LDA-based document models for ad-hoc retrieval. In *SIGIR*.