CS 386: Project 3 Report **Shivam Kumar** (160010010)
October 27, 2018

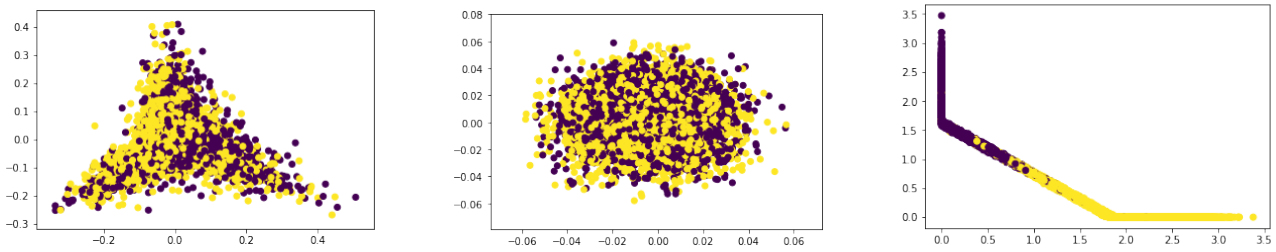
1 Introduction and Overview

It is easier to use numerical data (ND) in training machine learning (ML) models than using textual data (TD). That's because many ML algorithms work only on ND. LSA is one among the commonly used methods to convert TD into ND. One another method involves getting a numerical vector for each document (using methods like Bag of Words or TF-IDF) and applying dimensionality reduction technique like Autoencoders (AE).

In this report, we are going to discuss LSA and how it is used to convert TD into ND. Then we are going to compare LSA and AE in information retrieval (IR). After that, we are going to discuss the performance of LSA and Autoencoder with Logistic Regression (LG) and Neural Network(NN) in supervised learning.

2 Methods

- Removing Punctuation, Converting to Lowercase, and Removing Stopwords - A stop word is a common word found is almost every document. As a result, It contains almost no information to differentiate between documents. Same goes with punctuation and uppercasing also.
- Stemming and Lemmatization - Different forms of a word has similar meaning, so it is better to treat them same. Stemming tries to do this by removing suffixes. By doing so, It creates meaningless words also. Lemmatization fix this shortcoming by checking the validity of word before modifying it.
- TF-IDF (term frequency - inverse document frequency) - It reduces the weights of those words which are present in most of the documents. So that more differentiating words can have more priority.
- LSA (Latent semantic analysis) - It is used to assign a representation to each terms and documents in the k-dimensional space of k concepts. It assigns representations to terms and documents such that similar terms/documents have similar representations and different terms/documents have different representations. It uses matrix-decomposition using SVD (Singular value decomposition) to achieve this objective.
- AE - It is a NN that is trained to copy its input to output, in the process of training, it also learn useful features of input. We usually use these features to preform some other tasks like supervised learning or clustering etc.



(a) LSA (b) Autoencoder (c) Neural Network

Figure 1: These figures contain the visualizations of encodings obtained by different methods. Here, x-axis is first dimension of encoding and y-axis is second dimension. Different colors represent different classes.

3 Analyses of Results

After testing IR using LSA on several queries, we got relevant reviews for some queries (good, nice, etc.) but not so relevant for some others (price, delivery etc).

To compare LSA and AE in IR, we have to create some (Query(q),Review(r)) pairs manually where r is most relevant review for q. These pairs will be highly subjective i.e. for query q, r1 may seem most relevant to one person and r2 to some another person. Another problem is generating (q,r) pairs will be time consuming. So, we decided to compare both methods by using LG model to predict Recommendation ID (RID) given the encoding of text from LSA and AE.

After training LG on the encodings given by LSA and AE, we got 80% accuracy in both cases. After close inspection we found that, in both cases LG model is always predicting RID = 1 and RID was 1 in 80% of data. This may be case because of using binary cross entropy as our loss function to train our LG model and use of loss function like F1 loss, which also takes care of recall and precision, can improve accuracy. But after observing the plot of encoding of LSA and AE (in Figure 1), we couldn't find a boundary that can separate class-A (RID = 1) from class-B (RID = 0).

It is also possible that there isn't much information present in document to differentiate between reviews of class-A and class-B. To test that, we have created a simple NN having 3 hidden layer of 100, 10 and 2 units. We got around 90% accuracy when number of reviews of class-A and class-B was similar. After plotting the output of last layer, we also observed the separation between documents of different classes (in Figure 1). So, there is information present in document that can be used to differentiate between these two classes.

LSA and AE might be learning some features which can't be used to differentiate between documents of different classes that we have tested on. More experiments are needed to verify this claim. But we can say confidently form our experiment that, it is good to train our model once on TF-IDF matrix, without using LSA or AE to reduce dimension, when we are working with supervised learning. In this particular case, it gave better results.