CS 475/675 Project Proposal

Shihong Wei, Jingyi Ren, Xuan Wu, Zihui Wang
swei15, jren20, xwu78, zwang220

Abstract

Movie reviews reflect multiple elements of a movie and contain useful information to both industry and viewers. In this project, we will perform a sentiment analysis on textual movie reviews. We will train and develop different models that predict whether a textual movie review is positive or negative, and investigate what part of information or features are important. Furthermore, a thorough comparative analysis will be conducted both for different textual feature engineering techniques (such as word2vec, N-Gram, TFIDF) and for models from different classes.

1 Project choice

Choose either a **methods** or **applications** project, and a subarea from the below table.

☒ **Applications**				
☐ Genomics data	☐ Healthcare data	☒ Text data	☐ Image data	☐ Finance data
☐ **Methods**				
☐ Fairness in ML	☐ Interpretable ML	☐ Graphical Models	☐ Robust ML	☐ Privacy in ML

2 Introduction

There are thousands of movies coming out each year. Movie reviews reflect the quality of movies, and influence many people in their choice of movie-watching. In this project, we want to perform a thorough sentiment analysis to identify the polarity of textual reviews. This research work can hopefully help viewers decide whether to watch a newly released movie or not. It may also be of interest to the movie industry to tell what kind of movies the market will like and help recommend movies to users based on previous reviews.

Each input is a paragraph of movie review consisting of several English sentences. We will train and develop different SVM, neural network and LSTM models to predict whether the movie review is positive or negative. Then, with the trained models, we will investigate what are key features in a movie review that reveals most of its attitude (e.g., frequency of particular key words). Furthermore, we will conduct a comparison analysis in following two aspects: First, we will compare different texture feature engineering techniques such as word2vec, N-Gram, TFIDF. Second, we will compare those best performing models in each classes based not only on their test accuracy, time complexity, but also on their fairness and interpretability (details explained in the deliverables section).

3 Dataset and Features

The main dataset that we will be using in this project is the Large Movie Review Dataset (LMRD) from https://ai.stanford.edu/ amaas/-data/sentiment/. It is a collection of English textual movie reviews extracted before year 2011 from IMDB website, each with a label being either positive or negative (see, e.g., [1].). There are around 25,000 examples in the original training set, and around 25,000 examples in the original testing set, so we may further use a training (75%)-validation (15%)- testing (15%) split. For data pre-processing purpose, the following aspects will be considered: First, because each example is stored in a separate .txt document, we need to extract and read in all examples. Second, we will use standard NLP techniques such as tokenization, converting all words to lowercase, and removing stop-words, periods, question marks, punctuations, etc. In particular, we will consider using word2vec (see, e.g., [2]) to construct feature space. Besides, methods such as N-Gram and TFIDF, discussed in the course will also be applied to extract features. Furthermore, because the LMRD database does not include example information such as movie type, country, when evaluating the fairness of the model, we may try out web crawler algorithms on IMDB website to gather an extra testing set for movie reviews after year 2011.

An example with "negative" label from the LMRD dataset is as follows:"It looks to me as if the creators of The Class Of Nuke 'Em High wanted it to become a cult film, but it ends up as any old high school Bmovie, only tackier. The satire feels totally overshadowed by the extremely steretyped characters. It's very unfunny, even for a turkey."

4 Methods

We plan to use SVM and CNN by modifying and running the corresponding homework codes. For LSTM model, since it is very time-consuming (with high accuracy, though), we plan to use Keras deep learning library to create an LSTM model with a subset of the whole dataset (for example, choose the top 5000 words from the dataset). In addition, we plan to compare performance between these methods by using a logistic regression with bag-of-words features as the baseline.

For the SVM, the hypothesis class is linear functions, the loss function is the Hinge loss, and the optimization approach is the QP solver. SVM tries to finds the best margin that separates the classes, which reduces the risk of error on the data. Therefore, it works well with unstructured and semi-structured data like text and images.

For neural networks, the hypothesis class is the universal function approximator. Here, we use Convolutional Neural Network(CNN) and Long Short-Term Memory(LSTM) separately. For both of them, we use binary cross entropy as the loss function, and take Adam Optimizer as our optimization approach. In CNNs, by applying convolutions and pooling layers, we can extract relevant information(especially spatial relationships) from "images", which are matrices constructed by representing each word with a vector of numbers of a specific length and stacking a bunch of words on top of each other. LSTMs are a special kind of Recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems. LSTM introduces long-term memory into RNNs and deals with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs.[3]

5 Deliverables

5.1 Must accomplish

1. Appropriate data preprocessing: Correct tokenization of the raw textual data should be performed. Details are explained in the dataset section. In particular, for the N-Gram and TFIDF part, we will try out our own codes based on previous homeworks, and for the word2vec part we will learn and use the python library named "word2vec".
2. Valid modeling procedure and criteria: The standard training-validation-testing split should be followed to ensure fair comparison and to avoid information leakage. We will select different model structures within SVM, NN, or LSTM based on both accuracy and information criteria. Besides modifying the relevant homework codes, the libraries are stated in the methods section.
3. Exploration of model interpretation in textual analysis: For example, we want to explore what are the key words that appears most frequently in the movie review contexts, and how they will affects our trained model in classify an example.

5.2 Expect to accomplish

1. Optimal model parameters and hyperparameters (for example, the number of training epochs, the size of training batches, and the number of neurons in CNN and LSTM, and the γ of the RBF kernel, and the penalty C in SVM) should be chosen based on some grid search methods.
2. High test accuracy of models: With a relatively large-sized and potentially representative sample, we expect our trained model to perform well on the testing set under at least one textual feature engineering techniques among word2vec, TFIDF and N-Gram.
3. Model comparison across different classes: For the best performing SVM, NN, LSTM models, we want compare their relative test accuracy and rough time complexity, and we want to explain at least partially the reason behind it.

5.3 Would like to accomplish

1. Coding and optimizing relevant algorithms to improve computational efficiency and to avoid redundant intermediate computation and avoid being too time-consuming.
2. Detecting and handling potential overfitting problem when training the models. For example, plotting out the accuracy against the step size and trying out techniques such as regularization or dropout if possible.
3. Further evaluation of the model performance and fairness based on the domain of movie review. For example, we will examine whether the trained model is fair across long and short reviews. Moreover, we will learn and try out some web crawler algorithms to construct an extra testing data set on IMDB that includes information such as movie type and country. Then, we will examine whether our trained model is fair in dealing with the review texts of different types of movies.

References

[1] Maas, A.L. & Daly, R.E. & Pham, P.T. & Huang,D. & Ng, A.Y & Potts,C (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).* Available at https://ai.stanford.edu/ amaas/data/sentiment/

[2] Mikolov, T. & Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.