CS5350 - Final Project - Movie Review Classification

Overview

My submissions to Kaggle used the TFIDF and Roberta datasets.The algorithms I have used are Averaged Perceptron, Perceptron Bagging, Perceptron Adaboost, Support Vector Machine (SVM) and Logistic Regression. My project focused on evaluating the performance of different linear classifiers and the composition of them.

Important Ideas

The features were extracted by non-linear models. Also, linear classifiers are agnostic PAC-learnable because the VC-dimension is linear in the dimensionality. Hence, I thought it would be a good idea to apply linear classifiers on these features. I used Averaged Perceptron because it usually generalizes better than the original Perceptron. I used SVM because it is backed by the theoretical guarantee that maximizing margin tends to lower the generalization error. I also used Logistic Regression with regularization because it is a probabilistic formulation of a linear classifier, and I would like to compare its performance with other linear classifiers.

For non-linear models, I used Perceptron Bagging and Perceptron Adaboost. For Perceptron Bagging, there are multiple Averaged Perceptrons each trained on a subset of examples sampled with replacement, and each Perceptron has a vote at test time. For Perceptron Adaboost, the final hypothesis is a threshold function that takes in a linear combination of weak learners (Averaged Perceptrons). Perceptron Adaboost behaves like a two-layer threshold neural network, which can represent any boolean functions unlike linear classifiers. These ensemble methods can reduce variance and increase the complexity of the decision boundary, so they have the potential to generalize or train better than linear classifiers.

What I have Learned

I have learned mostly about the practical techniques of implementing machine learning algorithms. Since I am maintaining a large library of machine learning algorithms, I have to organize the code and group relevant code and data in the same folder. For example, I have a file called data_util.py that includes code for processing and saving data.

I have found that cross validation is crucial in increasing the train and test accuracies of the learner. For example, Logistic Regression has a train and test accuracies of 80% with the best hyperparameters. However, I found that with the wrong hyperparameters, Logistic Regression can only achieve train and test accuracies of 50%.

I have also learned to utilize learning curves for debugging. If a learner does not have a smooth and increasing learning curve of accuracy, the model either has a wrong implementation or the model can be improved by hyperparameters tuning.

Summary and Discussion of Results

I first used Averaged Perceptron on the TFIDF, Spacy-embedding and Roberta dataset to examine which feature leads to better performance. With the best hyperparameters, Averaged Perceptron performs the best on Roberta, with a training accuracy of 0.816 and a test accuracy of 0.806. Averaged Perceptron overfits on TFIDF, with a training accuracy of 0.939 and a test accuracy of 0.751. Averaged Perceptron underfits on Spacy-embedding, with a training accuracy of 0.77 and a testing accuracy of 0.738.

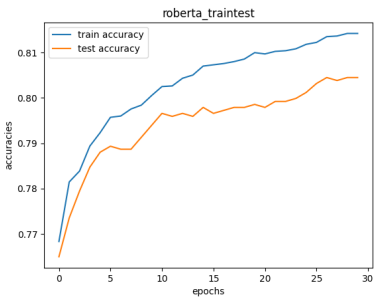

Figure 1 accuracies of Averaged Perceptron on Roberta dataset

Both SVM and Logistic Regression have similar training and testing accuracies to Averaged Perceptron on the Roberta dataset. However, the learning curves of SVM and Logistic Regression are not very smooth, probably because of Stochastic Gradient Descent. SVM tends to perform better with a large regularization coefficient C and small learning rate. In contrast,

Logistic Regression tends to perform better with small regularization coefficient 1/sigma**2 and a larger learning rate.

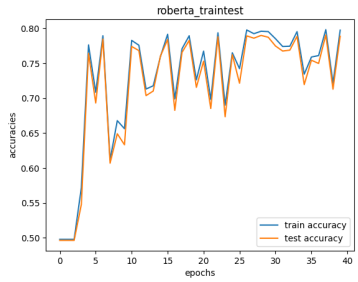

Figure 2 accuracies of SVM on Roberta dataset

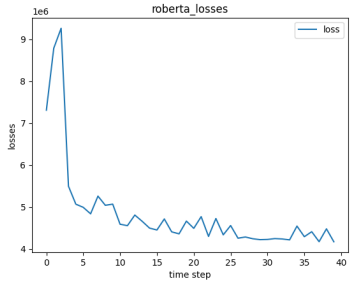

Figure 3 training losses of SVM

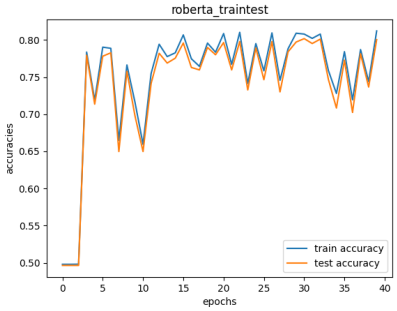

Figure 4 accuracies of Logistic Regression on Roberta dataset

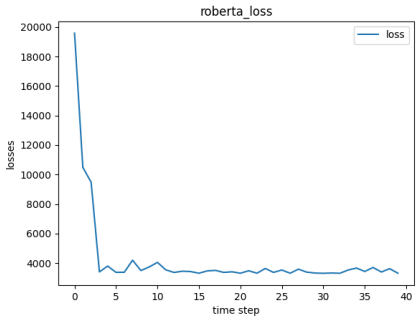

Figure 5 training losses of Logistic Regression

Perceptron Bagging does not outperform the other linear classifiers on the Roberta dataset. They have roughly the same training and testing accuracies. For Bagging, I used 40 Averaged Perceptron, each trained on half of the training data.

For Perceptron Adaboost, it performs the best out of all the models so far, especially for training accuracy. It is trained for 200 iterations. At each iteration, a weak learner (Averaged Perceptron) with weighted error less than 0.5 is generated by learning from a dataset sampled from the training data according to the distribution of Adaboost over examples. The training accuracy is 0.84 and the test accuracy is 0.82. The model is slightly overfitting.

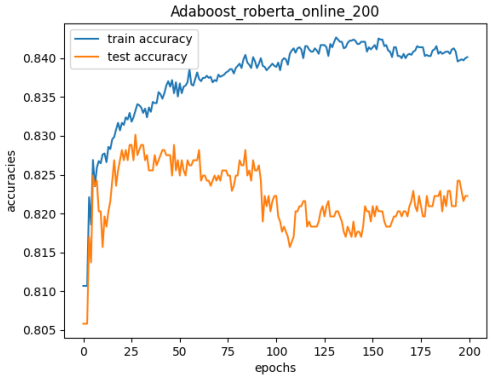

Figure 6 accuracies of Perceptron Adaboost on Roberta dataset

Future Work

I would like to develop a neural network that learns by comparing examples. It would take in two examples, compute a score for each of the examples. Then, it would compute a cross entropy loss on the two scores based on which example should have a higher score.

I would also like to have access to the original raw dataset of movie reviews, cluster the features extracted by Roberta and examine whether the features match the relationships of the original movie text reviews.