Sentiment Analysis

Marcus Tan

May 13, 2021

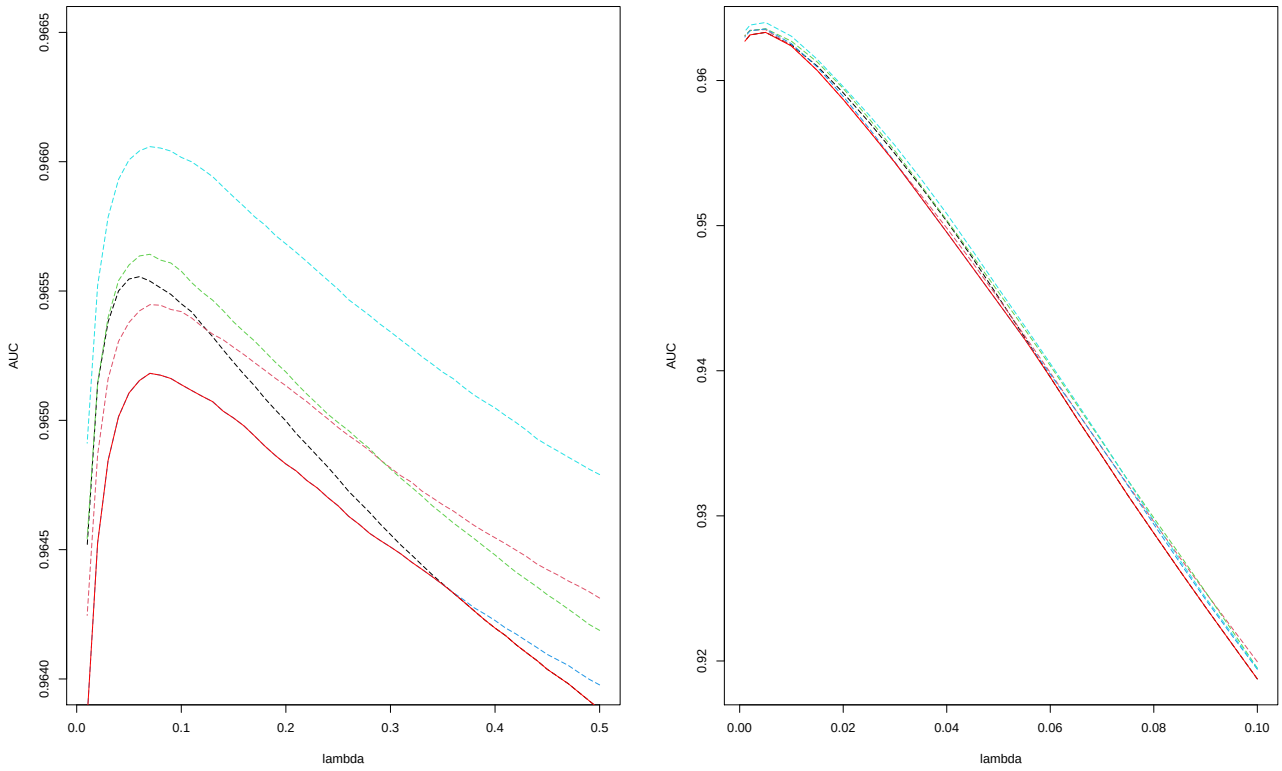
1 Overview

IMDB data consisting of reviews and corresponding sentiments (1 for positive and 0 for negative) were used to train models for predicting sentiments (outputs) given reviews (inputs). Five sets of splits with equal proportion of training and testing data were generated for cross validation and model parameter tuning. Each review was first tokenized into words and vectorized based on a list of 983 1- to 4-gram terms in the file "myvocab.txt." The generation of the list of terms have been described in a separate html document ("construct_vocab.html"). Each entry of a vector representing a review contained the number of occurrences of the terms in the review. When the vectors for all reviews were combined, they form a document-term matrix that can be used directly as the input for a model. The objective of this project is to train a model that produces an AUC ≥ 0.96 for all splits.

2 Model details

Two types of models — ElasticNet logistic regression (from the package `glmnet`) and Xgboost binary model— were explored. For the ElasticNet models, different alphas were experimented (0, 0.2 and 0.5). Alpha = 0 contains only the Ridge penalty and $0 <$ alpha < 1 is a mixture of Ridge and Lasso penalties. For each alpha, lambda was optimized on all splits by evaluating the model on a fixed sequence of lambda's and choosing the lambda that result in the maximum minimum AUC over all splits. The process can be understood with Figure 1 for alpha=0 and alpha=0.2, where the best lambda corresponds to the peak of the solid curve (min AUC over all splits). As lambda is sufficiently large, the AUC monotonically decreases with lambda. As alpha increases, the peak is shifted to the left and disappears for some alpha $\in (0.2, 0.5)$. The max min AUC's for alpha=0, 0.2 and 0.5 were found to be 0.9648, 0.9630 and 0.9627, respectively. This indicates a slight reduction in AUC as alpha is increased and that the ridge penalty results in the best ElasticNet model.

Xgboost models with different max numbers of boosting iterations (`nrounds`), max depths (`max.depth`) and learning rates (`eta`) were also explored. In general, AUC increased with larger `nrounds` and saturated when `nrounds` was sufficiently large (hundreds of iterations). `max.depth` and `eta` were manually tuned. The optimal `max.depth` appeared to be somewhere between 6 and 12 and the optimal learning rate was in the region 0.1 to 1. The best min AUC obtained was 0.9529 using `nrounds`=400, `max.depth`=10, `eta`=0.25. Running times were much longer than the ElasticNet model in general (averaging 51s vs 19s) so it was not used as the final model for submission.



(a) (b)

Figure 1: AUC as a function of Lasso penalty parameter lambda for training-test splits 1 to 5 (dashed curves) and the minimum AUC of all splits (solid curve) for (a) alpha=0 (Ridge penalty) and (b) alpha=0.2 (mixture of Ridge and Lasso penalties).

3 Model validation

A MacBook Pro with 2.2 GHz Quad-Core Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory was used to run the code. The performance of the best ElasticNet model (ridge penalty only with $\lambda = 0.07$) is shown in Table 1. Running times are listed in Table 2.

Table 1: AUC for a logistic regression model with ridge penalty and $\lambda = 0.07$ using the provided 5 test-training splits.

Split	1	2	3	4	5	Min
AUC	0.9661	0.9650	0.9655	0.9654	0.9648	0.9648

Table 2: Running times in seconds for a logistic regression model with ridge penalty and $\lambda = 0.07$ using the provided 5 test-training splits.

Split	1	2	3	4	5	Average
Elapsed time (s)	18.34	17.46	18.06	20.62	20.86	19.07

One shortcoming of the model can be inferred from the discrepancy between prediction and actual sentiment of Review 598 (first review of the Split 1 test data). The review is the following:

> "Now, Throw Momma from the Train was not a great comedy, but it is a load of fun and makes you laugh. The title may seem a little strange, but the entire movie isn't literally about that, although it is about something just as sinister. Danny De Vito basically wants to kill his overbearing mother, and

> fast forward a little bit, some random and funny events take place. The premise is quite funny, and the things that Billy Crystal and Danny De Vito get into were great. Some of the scenes seemed to not fit in for me, but this didn't make it a bad movie. For what it is, a wacky comedy, it pulls it off well and should be seen once just to say you saw it."

Low probability ($0.352 < 0.5$) was predicted by the model, which indicated a likely negative sentiment, though the actual sentiment is positive with a score of 7 out of 10. The discrepancy was due to the inability of the model to consider the interaction between different terms longer than the maximum size of the n-gram (in this case n=4). To see why, consider the list of terms appearing in the review: "although", "bad", "bad_movie", "basically", "bit", "but_this", "didn't", "entire", "fast_forward", "fun", "great", "just", "makes", "may", "of_fun", "off", "premise", "quite", "random", "seemed", "seemed_to", "should", "well."

In that review, the phrase "but this didn't make it a bad movie" was not taken to support a positive sentiment because "didn't" and "bad_movie" were separate terms in the vocab list and both indicated negative sentiment that decreased the predicted probability. The root cause of the failure is the inability to consider the ordering and interaction between terms. That is, if the order was "bad_movie" followed by "didn't"" or both terms are far apart, the chance of a negative sentiment would be high but would not indicate negative sentiment otherwise. We could increase n of the n-gram to include such phrase in the candidate terms but it will likely drop out of the final vocab list as such occurrence is rare and larger n without trimming the vocab results in sparser vector.

Another weakness is the use of auxiliary verbs like "just", "may", "seemed", "seemed_to", "should" etc as terms, which may not be a good indicator of positive or negative sentiment when used separately. Combining those verbs with actual positive or negative terms in a more complex model can potentially improve the accuracy by providing an additional measure of the degree of positivity or negativity. For example, the *paragraph vector* method proposed by Quoc Le and Tomas Mikolov can overcome the above issues by incorporating word ordering and semantics in the model.

4 Conclusion

An ElasticNet logistic regression with ridge penalty model was trained with a set of movie review data and was able to achieve an AUC of > 0.96 on 5 random train-test sets. Higher accuracy can potentially be achieved by iteratively reviewing the vocab list, excluding terms that are not good indicators of sentiments and repeating the term selection process. More complex models like the *paragraph vector* method can potentially be used to incorporate ordering, interaction between terms and semantics to improve the model.