Case study 2

Ross Quinn
Mark Makris

Introduction

This assignment is based on using machine learning techniques to classify the polarity of sentences in movie reviews taken from Rotten Tomatoes. To do this, a library of polarity labels for sentences were obtained from rotten tomatoes along with semantic analysis data of the review. In the hope of improving on the classification, data was obtained from Amazon's Mechanical Turk crowdsourcing platform, which will be used in a majority vote fashion on the test dataset. The crowdsourced dataset is not perfect however, as spammers and biased workers can skew the voting which would result in an incorrect classification. As such, the Dawid & Skene method is used to apply weights to the workers' votes so that the effect of these spammers and biased workers is lessened.

Part 1: Model training using gold-standard data

The gold-standard dataset, titled 'gold.csv', is comprised of data on 5000 sentences featured in Rotten Tomatoes reviews, along with the polarity of that review as a whole (positive=fresh and negative=rotten). The dataset contains 1202 columns with the first and last column being the identifier of the sentence and the polarity of the review respectively. The rest of the columns are features extracted using Latent Semantic Analysis (LSA).

The test dataset, titled 'test.csv', has the same layout as the gold standard dataset but contains 5428 rows instead of 5000. It is composed of different sentences to those found in the gold standard dataset.

For this part of the assignment, a sample taken from the gold-standard dataset was used as a training dataset in a decision tree classifier acting on the test dataset. The gold dataset was reduced to 1000 random rows, with identifiers of the sentences being extracted being saved so that the same sentences could be extracted from the mturk dataset later on..

The sample taken from the gold-standard dataset was saved as a separate csv file titled 'gold_sample.csv'.

With these two samples, the decision tree classification could now be performed. We used the built-in decision tree classifier available in the 'sklearn' package for Python,

which takes as the data input the 1200 LSA extracted feature columns from ‘gold_sample.csv’ and takes the polarity column from this file as the target input.

In order to judge the accuracy of the model, the F-score, accuracy, and prediction probabilities are calculated and written to a text file titled ‘train_gold.txt’

Part 2: Model training using crowdsourced data with majority voting

The crowdsourced dataset, titled ‘mturk.csv’ was much larger than the first two, featuring 27,746 rows and 1203 columns. This dataset is similar in layout to the ‘gold.csv’ file although the way the data was created is different. Instead of taking the polarity from the rotten tomatoes website, the polarity labels were created by workers using the Amazon Mechanical Turk crowdsourcing platform. The extra column in this case is the identifier for the worker that performed that classification. This file was also reduced by taking all the rows with the same sentence identifiers as the gold_sample dataset, writing the resulting dataset to a new csv file titled ‘mturk_sample.csv’. This file has many more rows than the gold_sample dataset as each sentence is annotated by multiple people.

In order to calculate the majority vote of the identified sentence we took the average vote from the annotators for each id. We had to create a list of unique ids and for each unique id we would see if the annotators had more positive or negative labels in order to see what the true polarity was. This created a smaller dataset of 1000 rows as there are 1000 unique ids.

This dataset is then used as the training dataset in a modified decision tree classifier that uses the majority vote of the workers to assign the most popular label choice among the workers to the sentence. If the result was neutral we would classify that as a positive.

As before, the F-score, accuracy, and prediction probabilities are calculated and written to a text file titled ‘train_mv.txt’. Sklearn has very simple functions that can build the decision tree, calculate the f-score, accuracy, and prediction probabilities for easy analysis.

Part 3: model training using crowdsourced data with Dawid & Skene method

The Dawid & Skene method comes from the paper ‘Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm’ by A. P. Dawid and A. M. Skene, 1979. This method allows the error rate of each contributor to a dataset to be calculated, which allows a weighting system to be created that lessens the impact of biased voters and spammers in a majority vote system such as the one used in the ‘train_mv.py’ program.

This process uses the same process of majority vote to calculate the true polarity of each unique id as done in the last part. From there we had to calculate how accurate each annotator was compared to the true polarity found in the majority vote. We then recalculate the majority vote taking into account how accurate each worker was so we know to give more weight to more accurate workers. That process is then repeated until convergence and we have a new more accurate dataset. As described in the linked paper, the method also calls for the multiplication of the rows of the confusion matrix by the marginal probabilities of each of the classes in the dataset. This step was deemed unnecessary for this assignment as the number of positive and negative reviews in the datasets is approximately equal.

This dataset is then used as the training dataset in a modified decision tree classifier that uses the Dawid & Skene method of assessing the accuracy of the workers to assign the most likely label choice among the workers to the sentence, giving more weight to more accurate workers.

Again, the F-score, accuracy, and prediction probabilities are calculated and written to a text file titled ‘train_ds.txt’.

Part 4: Data description and results comparison

The following parameters were taken from the outputted results text files of each model:

	f-score	accuracy
Gold	51.8%	52.8%
Majority vote	51.2%	51.4%
Dawid & Skene	53.4%	52.3%

The best performances we had were using the Dawid & Skene method on the mechanical turk dataset and the gold standard dataset for training. The Dawid & Skene model marked an improvement in both the f-score and the accuracy over the simple majority vote model and original set, indicating that the weighting on the workers is a useful way to deal with crowdsourced data. We also noticed that scores would improve after two iterations but not any further after that.

The gold standard would be expected to be the most accurate as it's the true polarity of the sentences as opposed to the estimated polarity according to the workers. However, with the Dawid & Skene model the majority vote system was almost as accurate as the gold standard, meaning it was close to being as accurate as reasonably possible.

The following is a table displaying the number of positive and negative reviews contained in the gold_sample and mturk_sample datasets

	POS	NEG	TOTAL
GOLD	490	510	1000
MTURK	2968	2631	5599
TOTAL	3458	3141	6599

The following is a table showing how the positive and negative labels are distributed between the two datasets.

	POS	NEG	TOTAL
GOLD	.074	.076	.152
MTURK	.450	.400	.848
TOTAL	.524	.476	

The fact that positive labels are the minority in the gold_sample dataset but make up the majority of the mturk_sample dataset may point to a positive bias in the workers. Of course, there is a chance that more workers were randomly assigned positive sentences than negative ones.

Avg workers per sentence: 5599/188 = 29.782
We see that each worker only annatates a few of the total sentences.

Conclusion

The decision tree classifier was not really able to classify the dataset in any of the models created. The fact that the classifier struggled even when using the gold-standard dataset as training data indicates that the SLA data itself is not very useful for classification. The similarity between the f-scores and accuracy between the train_gold and the train_mv models does indicate that the crowdsourced data isn't too dissimilar from the gold-standard in terms of label distribution. There was always a slight increase in f-score and accuracy between the simple majority model and the Dawid & Skene model. This could indicate that the Dawid & Skene method is an improvement over majority vote. If we had a different dataset to test we could try to see if the issue is more with the given data set or the algorithm, but with what we have Dawid & Skene was the best just not by that much.