Project Report

CS-595-01 : Interactive and Transparent Machine Learning

Active Learning With Rationales for Sentiment Analysis of Uber Ride Reviews

By Purvank Patel (A20380792)
Guided by: Prof. Mustafa Bilgic

Table of Contents

Task

In this project, I am using an active learning with rationales approach for the sentiment analysis of “uber ride reviews”. I am comparing two approaches called “learning with rationales” and “learning without rationales”. Also, I am comparing two active learning approaches “UNC-PC (uncertain-prefer-conflict)” and “UNC (vanilla uncertainty sampling)”.

Dataset

Data Source

For this project, I have scrapped data by myself, using python’s beautiful soup library. I have scrapped two websites.

1) https://www.consumeraffairs.com/travel/uber.html
2) https://www.sitejabber.com/reviews/www.uber.com#reviews

The original data has two columns, ‘ride review’ and ‘ride rating’. It has a shape of (1344,2).

Target Variable

In the original dataset, I have added a target variable called “sentiment” based on the “ride ratings”. If a ride has a rating of 3 or higher, I have labeled it as a positive, and if it has a rating lower than 3, I have labeled it as a negative.

Features

Using the “ride reviews”, I have created two datasets containing two different sets of features.

1) TfIdf Features: Using scikit-learn’s TfIdfVectorizer, I have created a dataset containing tf-idf values of the words. I have used ngram_range = (1,3), and min_df = 5 as a parameters for TfIdfVectorizer. It will create a dataset containing tf-idf values of unigrams, bigrams, and trigrams with a minimum frequency of 5.

2) Binary Features: Using scikit-learn’s CountVectorizer. I have created a dataset containing presence/absence of words as a features. I have used ngram_range = (1,3) and min_df = 5 as a parameters for CountVectorizer. It will create a features containing the 1/0 values of unigrams, bigrams, and trigrams with a minimum frequency of 5.

Data Size

1) Before Pre-processing and feature engineering: (1344,2)
2) After Pre-processing and feature engineering: (1344,1564)

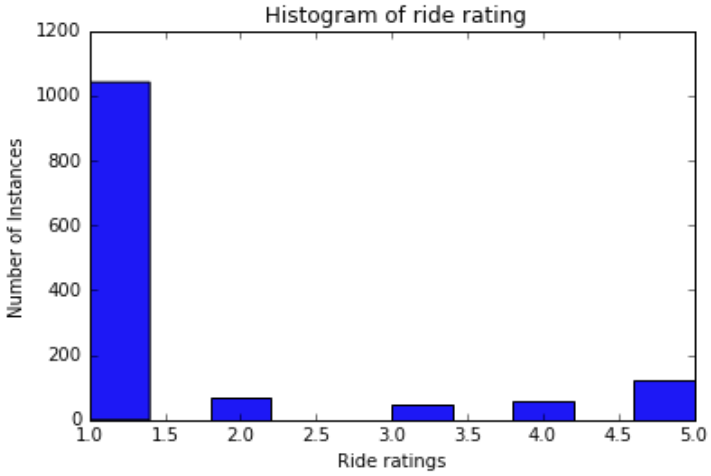
Preprocessing

I did some data-preprocessing on the ride reviews before using them to create features. I removed stop-words using, scikit learn's stopwords list, stanford's stopwords list, NLTK's stopwords list, and I have provided some of my own stopwords like "Uber" based on the ride reviews. After removing stopwords, I removed words who have the same stem, I have used NLTK's snowball stemmer to find out root of a word. I have kept the words in their original form, not in their stemmed form, I have only used stemming to remove words which have the same root like "bags" and "bag". Also, I have only used the alphabetic words, so I have removed phone numbers and words containing number in them from my text. Also, I have removed "ride rating" column from my data.

Visualization

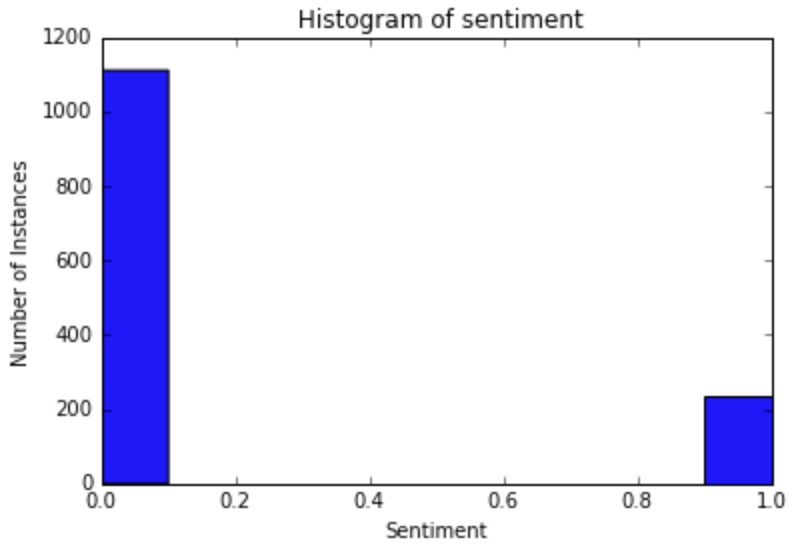
Feature Visualization

Below I have visualized the "ride rating" column from original dataset, as we can see that most of the reviews have a really low rating (< 3).



Target Visualization

As we can see from the above ride ratings histogram, most of the reviews have a very low ratings, which means in our dataset, most of the reviews have a negative sentiment. We can see that in below histogram of sentiment.



Approach

Learning with rationales

For learning with rationales, I have manually read 300 ride reviews and provided rationales for them. The rationales includes the words and phrases that support the sentiment of that particular ride. To provide rationales, I have built the GUI based on the Ipython Notebook, and I have saved those rationales into “rationales.csv” file.

These rationales are local, it means that every ride review has their own list of rationales, using these rationales I have matched them to the review’s features. If a feature is a rationale, then I have multiplied it’s value by 1, otherwise I have multiplied it’s value by 0.01.

```
My appointment time for auto repairs required a drop-off early am for it to be there and available for a mechanic to
begin the work. The Uber app was easily installed on my mobile device and was followed with the required information
being filled onto the blanks. When typing a phone number, it gave an order to check SMS texts for a confirmation code
to be used along with the number. No text sent with first try at 7:30 am. More tries, no texts and had to give up. Fo
ur hours later the texts with codes were sent at 11:08 am. Two attempts were unsuccessful at emailing Uber headquarte
rs in San Francisco that included a few lively suggestions. As follows; I suggest someone there unclog your ride summ
oning impacted website rectum by swallowing an online laxative, followed by a readily available Fleets enema from a p
harmacy.I now must wait 1 1/2 weeks to have repairs done from being a no-show at the shop. That means safety concerns
to drive as is. When using Uber once 9 months past, things went far better and a problem cleared up pronto by an apol
ogetic and most courteous driver. What a letdown on this go round. If the world's greatest ability is dependability,
why then are public service companies not aware of and learning such an important aspect of the business world? This
incident is a second of its kind with the same problem and still no fix. Then, unlike now, I had made contact with th
eir home base, only to be treated rudely and with great disrespect. Perhaps a reply is better not forthcoming after a
ll. I at least have a choice of 4 codes and might need one to request service 12 hours ahead of when it is needed. Ub
er folks, you had oughta uncrimp the do-do contained in the app from the Google store.


0


rationales
```

Active Learning

UNC

For UNC, I have selected the top 5 most uncertain instances based on the predicted probabilities, these are the top 5 instances that are closest to the decision boundary, and the classifier can't predict a label for these instances with certainty.
For these 5 instances, I provided rationales and then included them in the dataset.

UNC-PC

For UNC-PC, I have selected the top 20 most uncertain instances based on the predicted probabilities. I have created two list of rationales, positive class rationales and negative class rationales. The positive class rationales includes the rationales provided for positive class instances, and the negative class rationales includes the rationales provided for negative class instances.

For the 20 most uncertain instances, I have calculate the absolute difference of (no of words matching with the positive class rationales list - no of words matching with the negative class rationales list). Based on this difference, I have selected the top 5 instances which have lowest difference.

Below is a screenshot of active learning procedure.

```
----------50 instances----------------

---------------Active Learning----------------------

----------------Active Learning using UNC method--------------------------

Their P.O.S. survey vehicle causes $4600.00 damage to my truck.  They DO NOT have insurance.  So, you, Mr. Driver, ge
t your arse sued.  Welcome to UBER!  Michael Hung Thai Pham the subpoena is on it's way.  You WILL need a lawyer.  St
ill love Uber?  Ciao.


0


Do you want to change the label for this instance?: n


rationales: damage. sued. subpeona
```

Evaluation

Classifier

I have used Logistic Regression as a classifiers in my experiments. I have set the class_weight parameter to 'balanced' as my data has a class imbalance problem. Scikit learn will automatically adjust weights inversely proportional to class frequencies in the input data.

Performance Measure

I have used ROC curve as a performance measure, and based on the area under the curve (AUC), I have measured my classifier's performance.

Evaluation Strategy

Learning with rationales vs Learning without rationales

To compare, learning with rationales and learning without rationales. I have selected instances from dataset in the range from [50,300]. So, I first selected 50 instances as a training data, and rest of the instances as a testing data. Then I selected the 100 instances as a training data, and rest of the instances as a testing data. So, on. For learning with rationales, I have used a data containing rationales, and for learning without rationales, I have selected the data without the rationales.

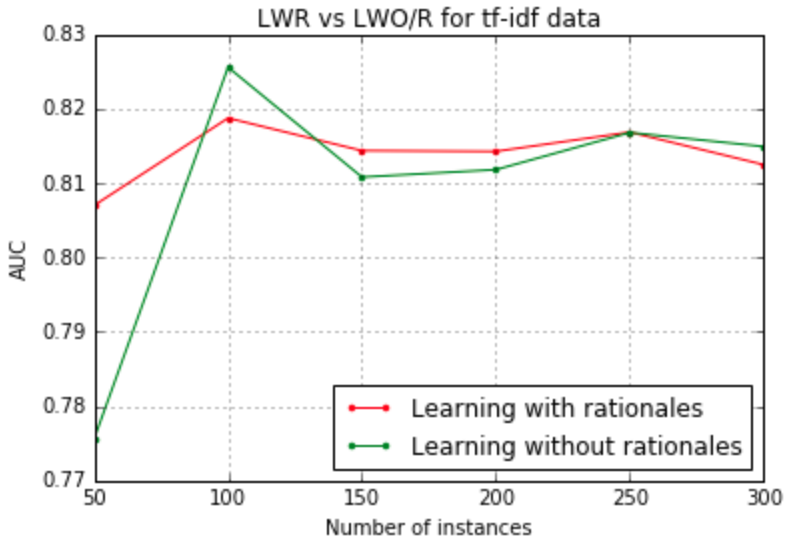
In the end, I have plot the learning curve to compare LWR and LWoR. Also, I have used both tf-idf dataset and binary dataset.

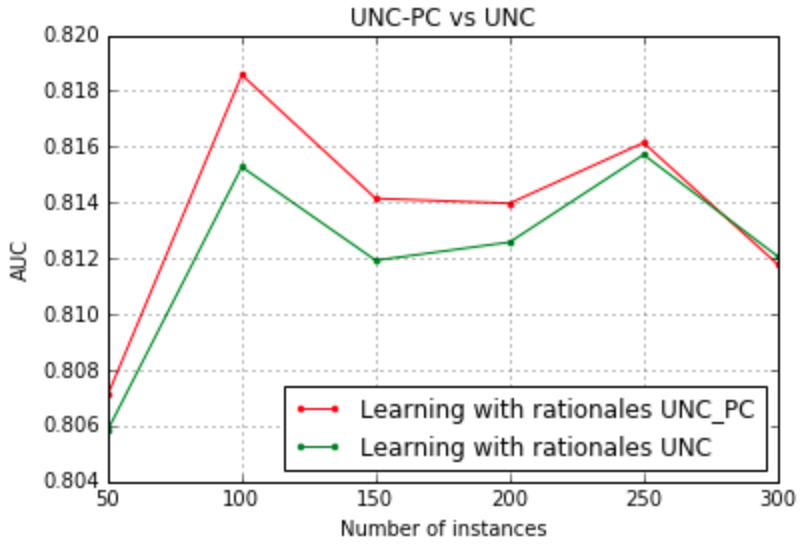
UNC-PC vs UNC

I have selected instances for query in both UNC-PC and UNC, as described above. And in the end, I have plotted the learning curve to compare them. I have used both tf-idf dataset and binary dataset for comparison.

Performance Results

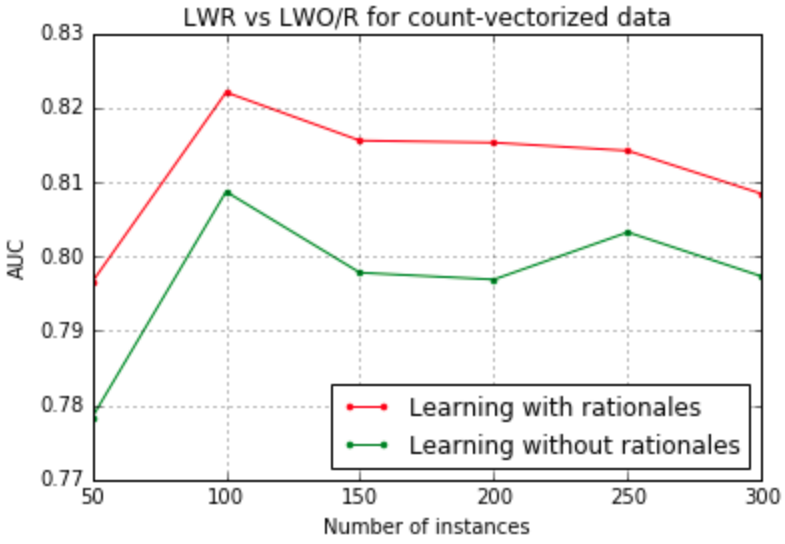
Performance on tf-idf dataset

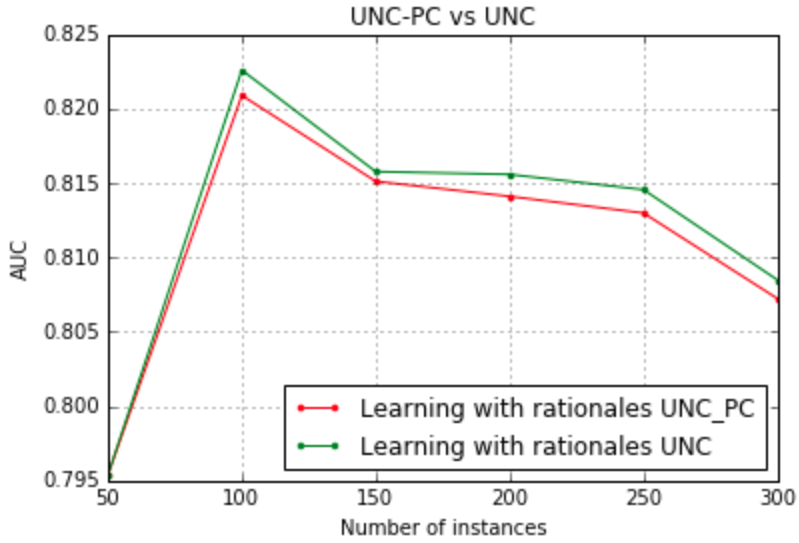




As we can see from the above learning curves, learning with rationales has a better AUC overall compared to learning with-out rationales. Also, UNC-PC has a better AUC compared to UNC.

Performance on binarized dataset





As we can see from the above graph, clearly learning with rationales has a better AUC compared to learning without rationales. Whereas, in UNC-PC the AUC is somewhat small compared to UNC, still there isn't that much difference between them.

Interesting/Unexpected Results

Some of the instances selected by UNC-PC and UNC for query made sense. For example, one of the instances selected for query was:

Review: first time I used Uber, they were great. The second time, they got lost trying to find me (I was at a public high school) and just gave up. They texted me to cancel. Didn't even try to work anything out or send another driver. I won't be using them again.

The above review had a positive sentiment, whereas the text of the reviews clearly shows that it expresses a negative sentiment.

Conclusion

- To conclude, learning with rationales works better than learning without rationales. Also, in some cases, UNC-PC performs better than UNC.
- Learning with rationales is a time-consuming but an effective approach to incorporate human feedback in the learning of a model.
- As seen in the interesting/unexpected results, Active Learning can select instances that are the model is the most uncertain about, and by querying those instances to an expert, expert can provide it's feedback. By incorporating, this rich feedback machine learning models can learn better, and they become more interactive and transparent.

References

- http://www.cs.iit.edu/~ml/pdfs/sharma-naaclhlt15.pdf
- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html