Homework 3

Jue Wang
jwr0983

Link to GitHub Repo:
https://github.com/MSIA/jwr0983_msia_text_analytics_2020/tree/homework3

Problem 1

The dataset I chose is the Amazon review dataset in the category of movies and TV's. The dataset contains the actual text of review and also the rating of the item by the reviewer. To shorten computation time, I subset the data to the first 500,000 reviews only.

Number of documents	500,000
Number of labels	5
Average word length in a review	164.2

Label distribution

LABEL	COUNT	PERC
1	30757	6.2 %
2	30068	6.0 %
3	59488	11.9 %
4	113145	22.6 %
5	266381	53.3 %

We see that there is some class imbalance within the dataset with ratings being quite right skewed towards higher ratings (4 and 5). Despite of this I kept this as a multi-class classification problem.

Problem 2

When building the logistic regression model and performing hyperparameter selection, I tried to experiment different sets of {bag-of-word, max_df, inverse of regularization strength C}. I calculated four performance metrics for model comparison, all of which are weighted by the class proportion to get a more sensible result. In terms of the choice of bag-of-word representation, I wanted to see whether using both 1 and 2-gram will increase prediction performance while not deteriorating run-time by too-much. In terms of max_df, I wanted to see whether we want to remove more words that appear too frequently, and how does that affect prediction accuracy. I also want to see how useful is C in terms of regularization and preventing overfitting.

The performance metrics overall are pretty similar among the four combinations I tried, but using unigram + bigram with max_df = 0.75 and C = 5 gives us the best performance across all

of the four metrics. We see that using 1+2 gram give us much more predictability compared to other cases.

	1-gram, max_df = 0.75, c = 5	1-gram, max_df = 0.95, c = 5	1+2-gram, max_df = 0.75, c = 5	1+2-gram, max_df = 0.95, c = 100
Micro-recall	0.6117893658895627	0.6117893658895627	0.6266300782437557	0.594884554458614
Micro-precision	0.573920399255954	0.573920399255954	0.6001078820434198	0.5777462460316921
Micro-F1	0.5838538093104109	0.5838538093104109	0.6087181377579953	0.5849339334408498
Accuracy	0.6117893658895627	0.6117893658895627	0.6266300782437557	0.594884554458614

Problem 3

When I was building the SVM model I tried to experiment with different combinations of {bag-of-word, Regularization parameter C, loss function}. I tried using unigram and unigram+bigram to see whether the influence of using more features would lead to the same effect as seen in logistic regression. With larger C the model is more prone to over-fitting, and we want to see the effect of that on the model performance as well. Using different loss function may lead the model to place weights of updates differently during optimization, so I also want to see the effect of that on the performance results.

	1-gram, c = 1, squared-hinge	1-gram, c = 1, hinge	12-gram, squared-hinge, c = 1	12-gram, squared-hinge, c = 10
Micro-recall	0.6132694369329728	0.615565547146263	0.6376626078051747	0.6218058466806407
Micro-precision	0.5721311660048944	0.5576933678565879	0.6049014953287122	0.5941675058683393
Micro-F1	0.5813251039890716	0.5543707865015062	0.6115646630450775	0.6025311263489262

Accuracy	0.6132694369329728	0.615565547146263	0.6376626078051747	0.6218058466806407

We see that the combination of 1+2-gram, squared-hinge loss and C = 1 gives us the best performing SVM model. We see that using squared-hinge or hinge loss is not affecting the performance metrics a lot, but using 1+2 gram gives us better prediction accuracy than using only 1-gram.

Problem 4

Example output:

“I love this movie”	{ "label":[5.0], "confidence":[[-1.6804154790883377, -1.476422895480371, -1.611500450055892, -1.2703161860222647, 1.4951135601852883]] }	“This is mediocre at best”	{ "label":[3.0], "confidence":[[-1.845461157806786, -0.31619098690410463, 0.052243176478961706, -2.0238628291537006, -0.8358914341238173]] }
“This is bad”	{ "label":[1.0], "confidence":[[0.311610988999506, -1.0533131660474546, -0.6906568701458538, -1.171976775691187, -0.6600008183880571]] }	“I really anticipated this movie and it did not disappoint me”	{ "label":[5.0], "confidence":[[-1.7547464132796176, -0.9195118809502818, -1.3139723083395982, 0.004078246274854314, 0.016270355619352278]] }
“It is ok but not the director's best work”	{ "label":[3.0], "confidence":[[-1.6664789911357905, -0.8158428132148654, 0.590326725364059, -1.23851984596658, -0.825338190795832]] }	"How can the leading actor be so bad"	{ "label":[1.0], "confidence":[[0.09865260505391471, -0.44400821340760566, -0.740048061242695, -0.719765548688714, -0.7458493256023809]] }