Statistical Natural Language Processing
Brandeis University
Fall 2015

PA2: Sentiment Classification - Max Entropy

Huilin Gang

October 13, 2015

1 Naive Bayes

I implemented Max Entropy Model with BagOfWords represents each document in my classifier. The converge rule in my model is if the accuracy between the current accuracy of dev dataset and the previous accuracy of dev dataset is nearly same(with different less than 0.01) and maintain for 100 times. Then the model is converged. I experimented with different size of training set and tested the performance of my model. The accuracy looks like below:

Table 1.1: Model Accuracy with different size of training set and number of features(with learning_rate = 0.001, and batch_size = 30)

training set	1000	10000	20000	50000	100000
dev set	100	1000	2000	5000	10000
test set	300	3000	6000	15000	30000
feature number	18959	82481	127400	227976	354324
accuracy	0.660	0.697	0.701	0.701	0.704

The accuracy is almost the same when floor into integer. So I planned to reduce the number of features in my model. Instead of using all the features in BagOfWords of training set, I tried to use the features which the presenting time is greater than the average times of each feature depends of the whole features.

Table 1.2: Model Accuracy with different size of training set and number of features(with learning_rate = 0.0001, and batch_size = 30)

training set	1000	10000	20000	50000	100000
dev set	100	1000	2000	5000	10000
test set	300	3000	6000	15000	30000
feature number	1540	4754	6694	9970	13355
accuracy	0.64	0.672	0.703	0.709	0.701

It is very important to know when to finish your learning progress of the model. Thus changing the converge rule will change the accuracy of test data set. However, it's very likely to be overfitting. The test accuracy could be greater than 70% if I change the converge rule. But it will take long time to train you model and upgrade the lambda.

Then I experimented with different learning_rate.

Table 1.3: Model Accuracy with different learning_rate(batch_size = 30)

learning_rate	1	0.1	0.01	0.001	0.0001
accuracy	65.7%	69.7%	68.4%	69.1%	65.2%

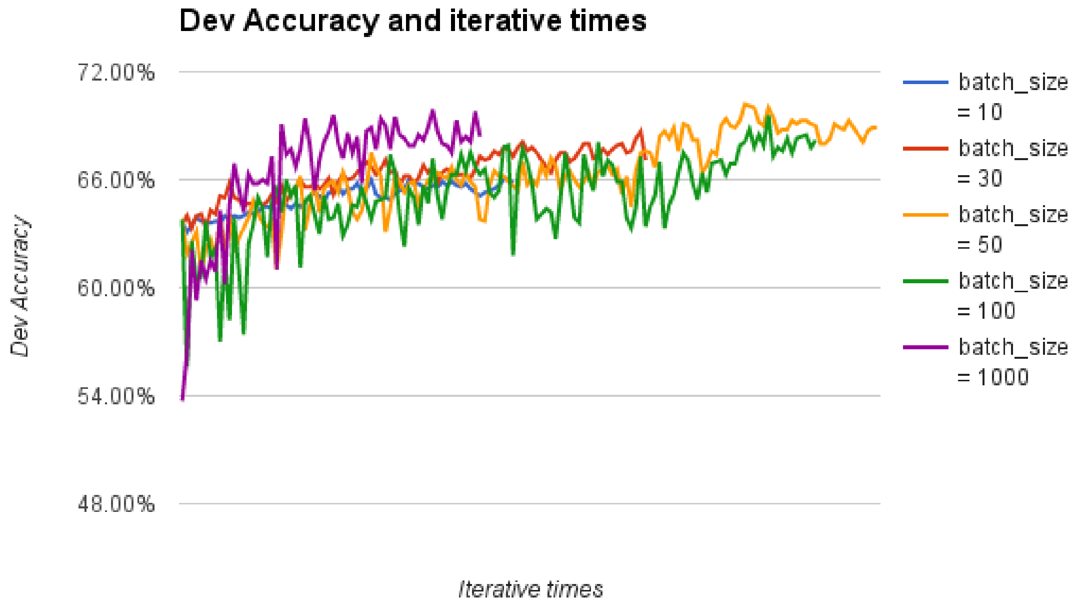
Changing the batch_size will also made the accuracy different.

Changing the batch size of max entropy model will have difference in dev data set accuracy. The chart looks like below:

With the increase of batch size, the dev accuracy is grow slightly.

Table 1.4: Model Accuracy with different batch_size(with learning_rate = 0.001)

batch_size	1	10	30	50	100	1000
iterate times	384	1765	4471	5782	4500	7
num of data points	384	17650	13413	10000	10000	10000
accuracy	0.614	0.658	0.671	0.675	0.711	0.680



2 Feature Engineering

For bag of word, I tried to eliminate all stop words and also the one with outlier frequency, but none of these works efficient in my model.

Table 2.1: Model Accuracy with feature engineering

Methods	Tokenization	Stemmed	NGram in sequence $[x_{i-2}, x_{i-1}, x_{i+1}, x_{i+1}]$
Feature numbers	2518	3664	30754
Accuracy	70.6%	70.1%	70.3%

3 Conclusion

It's of great fun to train my own Max Entropy Model with sentiment data set. I tried different feature engineering method and different range of the features. After implementing the max entropy model myself, I build a clearer mind on how the model works on real data set and what measures can we use to deal with real data set.