Evaluation

Fei Luo(fla414)

1. **Performance table (10 trials of 10-fold cross validation):**

Classifier type	Precision (macro-average)	Recall (macro-average)	F1 measure (macro-average)
bayes	**0.70185**	**0.81741**	**0.75520**
Best bayes	**0.79859**	**0.90136**	**0.86823**

For evaluation purpose, our classifier doesn't classify neutral documents. It's easy to modify the classifier to support neutral classification by setting an appropriate epsilon.

If P(positive) – P(negative) > epsilon: → positive
Else If P(negative) – P(positive) > epsilon: → negative
Else → neutral

2. **How do we improve the classifier?**
    1) We firstly implemented the basic naïve bayes classifier. The feature we selected is the term frequency. We used add-one smoothing. The overall performance is actually not bad.
    2) I tried the feature of term occurrence. The performance is similar with the first classifier.
    3) In addition to the unigram feature, I added bigram feature. However, the performance degraded significantly.
    4) I used the English stemmer from "nltk" module and found it significantly improved the overall performance.
    5) I tried the Laplacian smoothing, but it seemed that the performance didn't improve.
    6) I observed the precision of negative class and the recall of positive are the bottlenecks. This is caused by the imbalance of the number of positive and negative documents in the training data and it was getting worse in the case of bigram classifier, since the denominator

after adding the bigrams is increasing significantly. As Sara mentioned in class, I used the same number of positive and negative documents in the training process. This also improved the performance.

7) Finally, our best bayes classifier chooses the combination of unigram and bigram as features with stemmed words and removal of punctuation marks.

3. Our thoughts

1) One of the problem of the naïve bayes classifier using unigram as the feature is that it performs poorly in cases such as “This book is not good.”. At first, our naïve classifier classified this sentence as “positive”. So, adding the bigram or trigram features will definitely prevent this issue. Stemming is important since there are so many words appearing in different tenses and forms which in fact have the same sentiment.
2) For future improvement, we would try some feature selection algorithms, combination of unigram, bigram and trigram together, and some algorithms for handling negations.

4. Original results

```
9 th iteration
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
finish training classifier with bigram frequency
positive precision is:  0.986429657545
negative precision is:  0.610758196424
positive recall is:  0.850485611511
negative recall is:  0.952234432234
positive f measure is:  0.91335232238
negative f measure is:  0.743899386872
```