CSCI 3907/6907: Introduction to Statistical NLP

Assignment #3

Instructions:

- ✓ This assignment is due on **Friday November 9, 2018, by 11:59 pm**. Refer to course information about late submission policy. Late days are counted from 00:00 a.m. onwards.
- ✓ Submit your solution in a zipped folder through blackboard.
- ✓ All code must compile and run to receive full credit for coding parts.
- ✓ Include citations for any online resources used or group discussions.

Text Classification

In this assignment, you will use **scikit-learn**, a machine learning toolkit in Python, to implement text classifiers for sentiment analysis. Please read all instructions below carefully.

Datasets and evaluation:

You are given the following customer reviews dataset: `CR.zip`, which includes positive and negative reviews. CR is a small dataset that doesn't have train/test divisions, so you are required to evaluate the performance using **10-fold cross-validation**. Please use the following scikit-learn modules in your implementation:

scikit-learn documentation:

Bag-of-words (or ngrams) feature extraction using CountVectorizer:
http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Use **binary features** (1/0 rather than counts).

Naïve Bayes classifier:
http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

Logistic Regression classifier:
http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Cross validation:
http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

Classification report:
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

Question 1:
Using the scikit-learn modules described above, Implement the following models and report the performance (accuracy and F1) for the CR dataset:

a) [10 points] A Naïve Bayes classifier with add-1 smoothing using **binary** bag-of-words features.
b) [10 points] A Naïve Bayes classifier with add-1 smoothing using binary bag-of-ngrams features (with unigrams and bigrams).
c) [10 points] Logistic Regression classifier with L2 regularization (and default parameters) using binary bag-of-words features.
d) [10 points] Logistic Regression classifier with L2 regularization using binary bag-of-ngrams features (with unigrams and bigrams).

Performance report [10 points].

[optional bonus question – 10 points]

In this part, you are asked to implement a model that combines the advantages of generative and discriminative models: a logistic regression classifier with Naïve Bayes features.

In Naïve Bayes, we used the MLE probabilities of words/bigrams given a class label. We can use the ratio of these probabilities as features in logistic regression. Define the count vector for the positive class **p** as the smoothed sum of features for all instances that belong to that class:

$$p = 1 + \sum_{i:y(i)=1}^{\cdot} f(x_i)$$

where $f(x_i)$ is the binary feature vector for example x_i. Each element in p is the count for a specific word/bigram with add-1 smoothing. Similarly, the count vector for the negative class **q** is defined as:

$$q = 1 + \sum_{i:y(i)=0}^{\cdot} f(x_i)$$

The log-count ratio is defined as:

$$\mathbf{r} = \log\left(\frac{\mathbf{p}/||\mathbf{p}||_1}{\mathbf{q}/||\mathbf{q}||_1}\right)$$

which is the ratio of the positive to negative likelihoods for each word/bigram (the $||x||_1$ notation is the L_1 norm, which is the vector sum since all features are nonnegative). Now instead of using the binary feature vectors $f(x_i)$ as input to the logistic regression classifier, use the element-wise product of the log-count ratio vector and each feature vector: $r \odot f(x_i)$

Report the performance of this model on the CR dataset using the two sets of features: (a) bag-of-words only, and (b) bag-of-ngrams (unigrams and bigrams).