CS 180 MP 3: Naive Bayes

Marc Teves
2015-08007
THR

April 13, 2018

Requirements:

- Python 3
- `stemming` python module
- `scikit-learn` module
- `bash` or your favorite shell

Just run `main.sh` to start from the very beginning, or run `classify.sh` to classify based on the pre-computed feature vectors.

1. The Bernoulli Naive Bayes achieved a performance measure of around 0.9514. Refer to Figure 1.
2. The Multinomial Naive Bayes achieved a performance measure of around 0.9652. Refer to Figure 1.
3. Refer to Figure 2. As the lambda smoothing increased, the accuracy decreased along with it.
4. By removing common stop words and words that have less than 3 characters, the accuracy went up marginally for Bernoulli but went down marginally for Multinomial. Refer to Figure 3.
5. By making the dictionary smaller after reducing it to only stem words, accuracy worsened marginally for Bernoulli while it increased marginally for Multinomial. Refer to Figure 4.

```
reimu@toho-genso ~/acads/cs180/mp3 $ ./classify.sh
Bernoulli accuracy:
0.9513915591561808
Multinomial accuracy:
0.9652077062809107
```

Figure 1: Accuracy for regular dictionary extracted from the training set

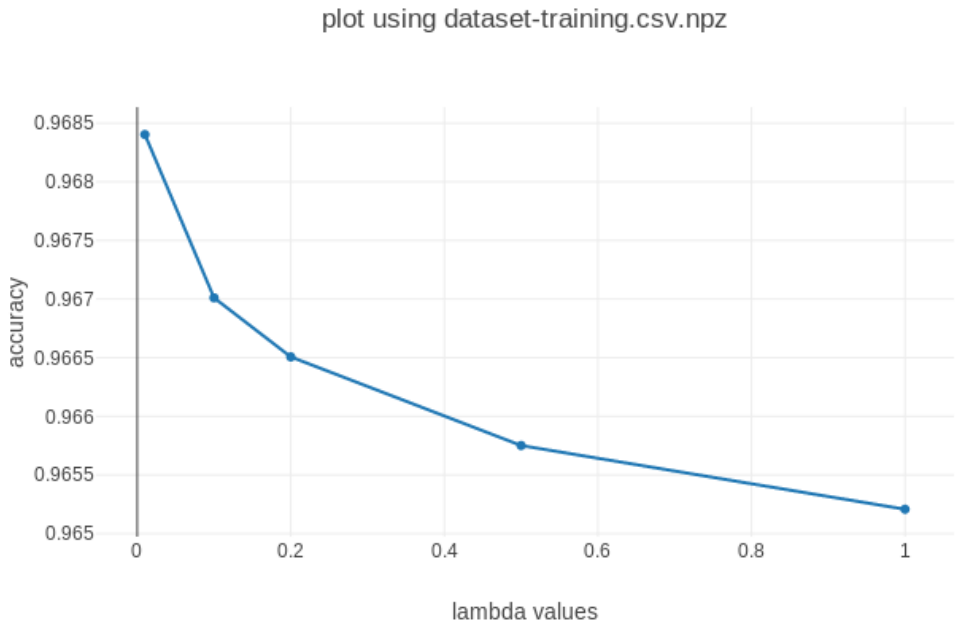

Figure 2: Plot of accuracy vs. lambda values. Accuracy decreases exponentially slower with higher lambda values.

```
Bernoulli accuracy:
0.9543748922685265
Multinomial accuracy:
0.9644917063339477
```

Figure 3: Accuracy for regular dictionary extracted from the training set, with stop words and very small words removed

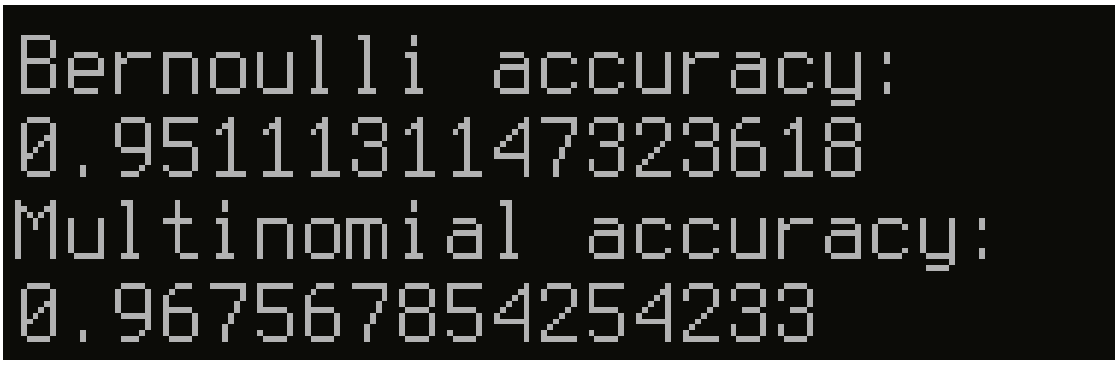

Figure 4: Accuracy for regular dictionary extracted from the training set, reduced to only stem words