CS187 - Homework 1, Part II

Lucas Freitas

February 13, 2013

12. Output from Python script (available in submission folder as $freitas\_hw1\_section3.py$):

```
Accuracy of prediction: 0.680000
Accuracy by class c:
    loc 0.506173
    hum 0.876923
    num 0.415929
    abbr 0.000000
    enty 0.648936
    desc 0.971014
```

13. Output from Python script (available in submission folder as $p6.py$):

```
Accuracy of prediction: 0.188000
Accuracy by class c:
    loc 0.000000
    hum 0.000000
    num 0.000000
    abbr 0.000000
    enty 1.000000
    desc 0.000000
```

Thus, the accuracy of the test set of a model that simply always guessed the most common class that occurs in the training set would be 0.188 for the training and test set given. Notice that the accuracy is far lower than the accuracy obtained using the Bernoulli method, and that the accuracy by class is 0 with the exception of the most common class. In general, the accuracy of that method for a generic training set with N classes, being i the most common class, and a test set with n entries and n_i entries that are of class i is:

Accuracy of prediction: $\frac{n_i}{N}$
Accuracy by class c: 1 if $c = i$, 0 otherwise.

14. Output from Python script (available in submission folder as $freitas\_hw1\_section4.py$):

    ```
    Accuracy of prediction: 0.752000
    Accuracy by class c:
        loc 0.802469
        hum 0.953846
        num 0.699115
        abbr 0.000000
        enty 0.638298
        desc 0.797101
    ```

15. (a) Higher accuracy in predicting the class labels of the test set: $freitas\_hw1\_section4.py$, the script that uses the Multinomial model.

    (b) Higher accuracy by class:

    i. loc: the Multinomial model
    ii. hum: the Multinomial model
    iii. num: the Multinomial model
    iv. abbr: same accuracy for both models (0%)
    v. entry: the Bernoulli model
    vi. desc: the Bernoulli model

16. The Bernoulli model would generate a higher class conditional probability for a very common word. For the Bernoulli method, $P(common\ word|\ any\ class) = 1$ for pretty much any class, since one occurrence of that very common word in the class would be enough to lead to a 1 probability.

    For the Multinomial model, on the other hand, $P(common\ word|\ any\ class) = 1$ only if every single word in the corpus of the class is that common word, since the Multinomial method considers the number of occurrences of the word in the text. Thus, the Bernoulli model would probably be the one which would generate a higher class conditional probability in that context.

17. Output from Python script (available in submission folder as $p5.py$):

    ```
    Accuracy of prediction: 0.798658
    Accuracy by class c:
        loc 0.795918
        hum 0.887850
        num 0.803419
        abbr 0.000000
        enty 0.823529
    ```

```
    desc 0.770833

Accuracy of prediction: 0.781879
Accuracy by class c:
    loc 0.891304
    hum 0.868217
    num 0.723214
    abbr 0.000000
    enty 0.770370
    desc 0.713115

Accuracy of prediction: 0.784874
Accuracy by class c:
    loc 0.787879
    hum 0.878788
    num 0.800000
    abbr 0.000000
    enty 0.734848
    desc 0.800000

Accuracy of prediction: 0.774790
Accuracy by class c:
    loc 0.870588
    hum 0.823529
    num 0.850000
    abbr 0.000000
    enty 0.654930
    desc 0.804348

Accuracy of prediction: 0.766387
Accuracy by class c:
    loc 0.804348
    hum 0.830769
    num 0.777778
    abbr 0.000000
    enty 0.740741
    desc 0.737589

Accuracy of prediction: 0.806723
Accuracy by class c:
    loc 0.759036
    hum 0.904762
    num 0.833333
    abbr 0.000000
    enty 0.810811
    desc 0.760684

Accuracy of prediction: 0.764706
Accuracy by class c:
    loc 0.811111
    hum 0.830645
    num 0.798246
    abbr 0.000000
    enty 0.700000
    desc 0.751938

Accuracy of prediction: 0.776471
Accuracy by class c:
    loc 0.797753
    hum 0.875000
    num 0.816514
    abbr 0.000000
    enty 0.719298
    desc 0.748148

Accuracy of prediction: 0.783193
Accuracy by class c:
    loc 0.816327
    hum 0.847826
    num 0.865979
    abbr 0.000000
    enty 0.693431
    desc 0.782609

Accuracy of prediction: 0.756303
Accuracy by class c:
    loc 0.822222
    hum 0.801587
    num 0.812500
    abbr 0.125000
    enty 0.730263
    desc 0.705036
```

18. Output from Python script (available in submission folder as $p7.py$):

```
Average accuracy of predictions: 0.780576
Accuracy by class c:
    loc 0.820367
    hum 0.853178
    num 0.809724
    abbr 0.025000
    enty 0.733879
    desc 0.763114
```

(a) Higher accuracy in predicting the class labels of the test set: the average of cross validation

(b) Higher accuracy by class:

i. loc: cross validation method

ii. hum: the Multinomial model

iii. num: cross validation method

iv. abbr: cross validation method

v. enty: cross validation method

vi. desc: the Multinomial model

As expected, cross-validation led to a higher accuracy than the Multinomial or Bernoulli method. That happens because in cross-validation, data is more randomly distributed, and every entry is used both for training and test at some point. Also, every single entry is used for test exactly once. That way, the training data is less likely to be biased, and actual patterns are more likely to be identified.