CS187 - Homework 1

Lucas Freitas

February 11, 2013

1. Using the formula given, we calculate that:

$$\hat{Pr}(Bush) = \frac{\#(Bush)}{N} = \frac{3}{5} = 0.6$$

$$\hat{Pr}(Obama) = \frac{\#(Obama)}{N} = \frac{2}{5} = 0.4$$

2. Output from Python script (available in submission folder as $p1.py$):

```
Pr(Bush) = 0.600000
Pr(Obama) = 0.400000
```

3. When calculating unsmoothed MLE estimates for the class priors, we only consider existent classes, which means that no matter which prior we are calculating, the numerator of the probability will always be 1 or greater. Class conditional probabilities of word types, and the unsmoothed estimates for the conditional probabilities, however, can be calculated as 0 if the word type is not included in the training set, which would lead to zero probabilities, and over-confidence on the MPA of a speech.

   Some other probabilities might also be extremely high depending on the training data, so there are very high bumps in the probabilities calculated without smoothing. By smoothing the probabilities, we avoid very high (close to 1) and very small (close or equal to 0) probabilities, which makes estimations more reasonable and fair, not causing over or under-confidence solely based on bias caused by training data. Therefore, smoothing leads to more precise class estimates, which makes it very important for the following problems.

4. Table with the data generated by the Python script (available in submission folder as $p2.py$):

Probability	Value
$\hat{Pr}$(terrorist|Bush)	0.800000
$\hat{Pr}$(freedom|Bush)	0.800000
$\hat{Pr}$(guided|Bush)	0.600000
$\hat{Pr}$(offensive|Bush)	0.800000
$\hat{Pr}$(amnesty|Bush)	0.800000
$\hat{Pr}$(honor|Bush)	0.800000
$\hat{Pr}$(mortgage|Bush)	0.200000
$\hat{Pr}$(terrorist|Obama)	0.750000
$\hat{Pr}$(freedom|Obama)	0.750000
$\hat{Pr}$(guided|Obama)	0.250000
$\hat{Pr}$(offensive|Obama)	0.250000
$\hat{Pr}$(amnesty|Obama)	0.250000
$\hat{Pr}$(honor|Obama)	0.750000
$\hat{Pr}$(mortgage|Obama)	0.750000

5. Since the maximum a posteriori (MAP) probability is a product of probabilities, which are all lower than 1, its value is usually way lower than 1. That makes it hard to differentiate probability values of two or more classes.

   As an illustration, which is smaller: 0.000001214 or 0.0000153? Much easier to see that -13.62 is smaller than -11.09, huh? See, all we had to do was calculate the log of the values! That way, it is better to take the logarithm of the a posteriori probability, which is going to be a negative number, but its module is greater than 1 at least, and differences between very small numbers are much easier to notice.

   Not only that, in certain languages like C, if we make the probability too small, the precision of a float or double will not be enough to hold a precise value for the final probability, and the probability can be truncated to zero. That way, by using logs, we can also avoid problems like that.

6. Output from Python script (available in submission folder as $p3.py$):

```
log(P(Bush|2008 speech)) = -6.519396
log(P(Obama|2008 speech)) = -4.028678
c_MAP = -4.028678

2008 speech prediction: Obama

log(P(Bush|2012 speech)) = -9.697449
```

```
log(P(Obama|2012 speech)) = -5.127290
c_MAP = -5.127290

2012 speech prediction: Obama
```

7. Output from Python script (available in submission folder as $freitas\_hw1\_section1.py$):

```
Accuracy of prediction: 0.500000
Accuracy by class c:
    bush 0.000000
    obama 1.000000
```

8. Table with the data generated by the Python script (available in submission folder as $p4.py$):

Probability	Value
$\hat{Pr}$(terrorist|Bush)	0.341667
$\hat{Pr}$(freedom|Bush)	0.350000
$\hat{Pr}$(guided|Bush)	0.025000
$\hat{Pr}$(offensive|Bush)	0.075000
$\hat{Pr}$(amnesty|Bush)	0.033333
$\hat{Pr}$(honor|Bush)	0.166667
$\hat{Pr}$(mortgage|Bush)	0.008333
$\hat{Pr}$(terrorist|Obama)	0.250000
$\hat{Pr}$(freedom|Obama)	0.150000
$\hat{Pr}$(guided|Obama)	0.050000
$\hat{Pr}$(offensive|Obama)	0.050000
$\hat{Pr}$(amnesty|Obama)	0.050000
$\hat{Pr}$(honor|Obama)	0.150000
$\hat{Pr}$(mortgage|Obama)	0.300000

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

```
log(P(Bush|2008 speech)) = -43.940531
log(P(Obama|2008 speech)) = -53.057878
c_MAP = -43.940531

2008 speech prediction: Bush

log(P(Bush|2012 speech)) = -25.498106
log(P(Obama|2012 speech)) = -8.833275
```

```
c_MAP = -8.833275

2012 speech prediction: Obama
```

10. Output from Python script (available in submission folder as $freitas\_hw1\_section2.py$):

```
Accuracy of prediction: 1.000000
Accuracy by class c:
    bush 1.000000
    obama 1.000000
```

11. (a) Higher accuracy in predicting the class labels of the test set: $freitas\_hw1\_section2.py$, the script that uses the Multinomial model.

    (b) Higher accuracy by class:

        i. Bush: $freitas\_hw1\_section2.py$, the script that uses the Multinomial model.

        ii. Obama: Both models have the same accuracy (100%).