# Naive Bayes Tutorial: Machine Learning (COMS3007) (Lab Session 3)

Devon Jarvis: 1365149

March 6, 2018

1. (a)

Table 1: Confusion Matrix for Question 1a

| | | Actual | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| Predicted | **Positive** | 1 | 2 |
| | **Negative** | 2 | 1 |

(b) The reviews I used were: ”The food is disgusting” and ”I will come back as soon as possible”. The reviews confused the model as it hadn’t seen many of the words in the reviews and as a result could predict the probability of each category with less accuracy as it had no prior knowledge to use to quantify the probability associated with the new words. Thus the output of the model becomes less calculated and more random in nature and the model will misclassify the reviews more regularly as a consequence. Further, due to the small amount of training data, the words the model had seen were largely misrepresented with probabilities associated to the words that are not completely accurate and thus the model may interpret some of the words I use as negative when I do not mean them to be in my reviews.

(c) There is a division my zero error. Meaning that the model must have seen a word it believed had zero positive likelihood and another word with zero negative probability. Resulting in the normalization element P(x) in the denominator of the classification formula being zero, as the entire solution space of the problem has a lieklihood of zero. This defies one of the main laws of statics which states that the likelihood of the entire solution space is 1 (ie: something must happen). The fact that this doesn’t happen with Laplace smoothing is due to the fact that with Laplace smoothing there is never a multiplication of 0 when calculating the prior probabilities. Thus new words will not give the prior probabilities of both or one of the categories a value of 0, as would happen without Laplace smoothing. This means that with Laplace smoothing the model is more robust to new words and relies more on the priors it does have when categorizing the reviews.

(d) The accuracy of the model improves and the model makes incorrect classifications less frequently. Removing stop words seems to reduce the clutter in the data the model uses (stop words often appear in both negative and positive reviews and thus don't aid the model in making its classifications). Thus the removal of such words leaves the model with data that it can use more objectively with most words it uses in its classifications providing material information to the model. The new confusion matrix was as follows:

Table 2: Confusion Matrix for Question 1d

| | | Actual | |
|---|---|---|---|
| Predicted | | **Positive** | **Negative** |
| | **Positive** | 2 | 1 |
| | **Negative** | 2 | 1 |

Table 3: Confusion Matrix for Question 2 Pang and Lee Data Set

| | | | Actual | |
|---|---|---|---|---|
| | | | **Positive** | **Negative** |
| 2. | Predicted | **Positive** | 90 | 23 |
| | | **Negative** | 15 | 72 |

With this data set there are a lot more words that occur very infrequently. Thus when the model tries to classify new reviews it mutliplies many very small probabitlities together and thus the likelihood used in the classification formula will become 0. Thus the model will find zero probabilities for the entire solution space ( which is impossible as the probabilitiy of the solution space is 1). Thus the classification accuracy will be very poor as the model will not be able to discern which posterior probability for the different groups (positive or negative review) is larger and will thus classify the review in a random category. To address this problem I removed all words where the positive and the negative probabilities from training had a less than 6% difference. Thus slimming down the amount of probabilities multiplied into the likelihood calculations, but not removing very material or informative probabilities from the calculation. The results are shown above, these results are reasonable as the model still has descriptive data to use to justify its new classifications, however much of the clutter of the large sparse data set used to train has been removed to help the model in its classifications. Thus a good performance (an 81% accuracy) as shown above is reasonable. It is however unlikely the model will outperform this as it still hasn't seen many of the new words that it may be tested on, and some of the words its been trained on may be incorrectly biased or have unrealistic probabilities. Thus more training data would help the model, espacially with the method of removing immaterial words from th likelihood calculation.

3. Proof for Mean ($\hat{\mu}$):
We need to solve the following maximization problem where $l$ denotes the log-likelihood of the normal distribution:

$$max_{\mu}\sigma^2 l(\mu, \sigma^2, x_1, ..., x_n)$$

The first order conditions for a maximum are

$$\frac{\delta}{\delta\mu} l(\mu, \sigma^2, x_1, ..., x_n) = 0$$
$$\frac{\delta}{\delta\sigma^2} l(\mu, \sigma^2, x_1, ..., x_n) = 0$$

The partial derivative of the log-likelihood with respect to the mean is

$$\frac{\delta}{\delta\mu} l(\mu, \sigma^2, x_1, ..., x_n)$$
$$= \frac{\delta}{\delta\mu}(-\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2\sigma^2}\Sigma_{j=1}^{n}(x_j - \mu)^2)$$
$$= \frac{1}{\sigma^2}\Sigma_{j=1}^{n}(x_j - \mu)$$
$$= \frac{1}{\sigma^2}(\Sigma_{j=1}^{n}x_j - n\mu)$$

Which is equal to 0 only if

$$\Sigma_{j=1}^{n}x_j - n\mu = 0$$

Therefore, the first of the two first-order conditions implies

$$\mu = \frac{1}{n}\Sigma_{j=1}^{n}x_j$$

Proof for Variance ($\hat{\sigma}^2$):
The Partial derivative of the log-likelihood with respect to the variance is

$$\frac{\delta}{\delta\sigma^2} l(\mu, \sigma^2, x_1, ..., x_n)$$
$$= \frac{\delta}{\delta\sigma^2}(-\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2\sigma^2}\Sigma_{j=1}^{n}(x_j - \mu)^2)$$
$$-\frac{n}{2\sigma^2} - [\frac{1}{2}\Sigma_{j=1}^{n}(x_j - \mu)^2]\frac{d}{d\sigma^2}(\frac{1}{\sigma^2})$$
$$-\frac{n}{2\sigma^2} - [\frac{1}{2}\Sigma_{j=1}^{n}(x_j - \mu)^2](-\frac{1}{(\sigma^2)^2}$$
$$-\frac{n}{2\sigma^2} + [\frac{1}{2}\Sigma_{j=1}^{n}(x_j - \mu)^2](\frac{1}{(\sigma^2)^2}$$
$$= \frac{1}{2\sigma^2}[\frac{1}{\sigma^2}\Sigma_{j=1}^{n}(x_j - \mu)^2 - n]$$

which, if we rule out $\sigma^2 = 0$, is equal to zero only if

$$\sigma^2 = \frac{1}{n}\Sigma_{j=1}^{n}(x_j - \mu)^2$$

Thus, the system of first order conditions is solved by

$$\mu = \frac{1}{n}\Sigma_{j=1}^{n}x_j$$
$$\sigma^2 = \frac{1}{n}\Sigma_{j=1}^{n}(x_j - \mu)^2$$

4. Yes the results are reasonable with the diagonal elements of the matrix holding the highest values in the table. Thus indicating that the model correctly classifies images more often than not. Naturally these results are statistical in nature and thus the model does not classify all the images correctly (this would be more of an indication of unreasonable results than even poor performace). A possible approach to improving results would, as always, to get more training data. Further, generating new data by performing affine transformations on the current data would help make the model more robust to many of the variances it may see when classifying new handwritten digits such as rotations and skew. Thus training on transformations of the trainng data would also help. The confusion matrix

follows on the next page.

Table 4: Confusion Matrix for Question 2 Pang and Lee Data Set

Actual

| | | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Predicted | **0** | 61 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| | **1** | 0 | 56 | 7 | 0 | 0 | 2 | 0 | 0 | 11 | 4 |
| | **2** | 2 | 2 | 76 | 5 | 0 | 0 | 0 | 0 | 3 | 0 |
| | **3** | 0 | 1 | 3 | 70 | 0 | 2 | 0 | 5 | 1 | 2 |
| | **4** | 2 | 3 | 0 | 0 | 72 | 0 | 0 | 1 | 1 | 0 |
| | **5** | 0 | 0 | 0 | 0 | 0 | 62 | 0 | 0 | 1 | 5 |
| | **6** | 2 | 1 | 0 | 0 | 0 | 2 | 82 | 0 | 0 | 0 |
| | **7** | 3 | 1 | 0 | 0 | 0 | 2 | 0 | 76 | 1 | 0 |
| | **8** | 0 | 10 | 2 | 1 | 0 | 2 | 0 | 2 | 62 | 9 |
| | **9** | 1 | 4 | 1 | 2 | 1 | 2 | 0 | 4 | 2 | 63 |