Naïve Bayes Tutorial
COMS3007

Benjamin Rosman / Phumlani Khoza

February 25, 2018

Due date: Monday, 5 March at 2.00pm.

Use your notes and any resources you find online to answer the following questions. Submit a report including the answers to the questions, and any code needed to generate the solutions.

1. We will look at using Naïve Bayes (NB) to do sentiment analysis – determining if some text has positive or negative sentiment. The file **simple-food-reviews.txt** contains 18 simple reviews for a restaurant. Each line of the fine starts with either a 1 or a -1, representing a positive or negative label.

   (a) Train a NB model on a random 12 reviews, **with** Laplace smoothing. Test on the remaining 6. Report the confusion matrix of your results.

   (b) Train the model on the full review set. Generate your own positive and negative reviews (1 of each) that can confuse the NB. Why did they confuse it?

   (c) Repeat step 1(a) **without** Laplace smoothing. Report your results. What do you notice?

   (d) A common trick in text processing is to remove "stop words". These are short, commonly-occurring words, such as "the", "and", "is", "a". Try removing some simple stop words from the data. For this case remove all words of length 1 or 2. How does this affect classifier performance?

2. Now try this on a real (but small) dataset. The file **movie-pang02.zip** contains a copy of Pang and Lee's Movie Review dataset. Train your NB classifier on this real dataset. Produce a confusion matrix of the results. Do you notice any extra challenges using this larger dataset? What can you do to address those? Use a random 90% of the data for training and the rest for testing. Are these results reasonable? How might you improve them?

3. Derive the parameters of the 1D Gaussian distribution (μ and σ^2) for maximum likelihood estimation (MLE) as was discussed in class.

4. The file **smalldigits.csv** contains 1797 8x8 images of hand written digits (modified from the sklearn digits dataset). Note the data here is low resolution and has been binarised for this example. Each row of the file has 65 elements: the first 64 are 0/1 values indicating the pixel value, and the last element is the class number (0-9). To visualise the nth digit, reshape the nth row into an 8x8 matrix. Train a NB classifier on a random 80% of the data, and use the remaining 20% to generate a confusion matrix showing its performance. Are these results reasonable? How might you improve them?