Homework 3
MLE, MAP, Bayes Rule, Naïve Bayes

CMU 10-601: Machine Learning (Fall 2017)
https://piazza.com/cmu/fall2017/10601b/
OUT: September 08, 2017
DUE: September 18, 2017 11:59 PM
Authors: Eti Rastogi, Sriram Kollipara, Oliver Liu

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only. See the Academic Integrity Section on piazza for more information: https://piazza.com/cmu/fall2017/10601b/home
- **Late Submission Policy:** See the late submission policy here: https://piazza.com/cmu/fall2017/10601b/home
- **Submitting your work:** This assignment will consist of two parts. Your solutions for Part 1 should be submitted to Gradescope, and your solutions for Part 2 should be submitted to Autolab.
  - **Gradescope:** For this assignment, we will use an online system called Gradescope for short answer and multiple choice questions. You can access the site here: https://gradescope.com/. You should already have been added to Gradescope using your Andrew ID. **Students are required to sign up using their Andrew ids or else they may not receive credit for all of their work.** If for some reason you have not been added to Gradescope, the entry code for this course is **97Y649**. Solutions can be handwritten, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Regrade requests can be made, however this gives the TA the right to regrade your entire paper, meaning if additional mistakes are found then points will be deducted.
  - **Autolab:** You can access the 10601 course on autolab by going to https://autolab.andrew.cmu.edu/ All programming assignments will be graded automatically on Autolab using Octave 3.8.2 and Python 2.7. You may develop your code in your favorite IDE, but please make sure that it runs as expected on Octave 3.8.2 or Python 2.7 before submitting. The code which you write will be executed remotely against a suite of tests, and the results are used to automatically assign you a grade. To make sure your code executes correctly on our servers, you should avoid using libraries which are not present in the basic Octave install. For Python users, you are encouraged to use the `numpy` package. The version of `numpy` used on Autolab is 1.7.1. We allow a maximum of 30 submissions per assignment, however it is likely that you will not need that many to complete this assignment. The deadline displayed on Autolab may not correspond to the actual deadline for this homework, since we are allowing late submissions (as discussed in the late submission policy on the course site).

Part 1: Submission Instructions [38 points]

To submit Part 1 of your work to Gradescope you may **do one of the following:**

- Print out ”Part 1: Multiple Choice and Short Answer Questions” (pages 3-8 of this handout) and write down your answers in the grey textboxes by hand. If you choose this option, you should only submit ”Part 1: Multiple Choice and Short Answer Questions” (pages 3-8) to Gradescope. Do not submit the entire Homework 2 Handout. Submissions must be made in PDF format.

  OR

- Use the provided .tex template to fill in your answers under the `%Your solution here` comment within each `tcolorbox` section. To submit your work for Part 1, you should submit your `tex` file with your solutions as a single `pdf` file and submit the file to Gradescope. Submissions must be made in PDF format.

Important Note: The final set of questions in Part 1 are based on Part 2 of this homework and should not be attempted until you have finished Part 2 first.

Part 1: Multiple Choice and Short Answer Questions [38 points]

1. [**1 pt**] **True or False**: One reason that the MAP might be preferred over the MLE is that MLE can have a tendancy to overfit small amounts of data.

2. [**2 pt**] Let X be the result of a coin toss, where $X = 1$ if it comes up heads and $X = 0$ otherwise. The coin has an unknown probability p_1 of coming up heads.
   Suppose that we observe the following sequence of of coin toss outcomes:

   $$(1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)$$

   What is the maximum likelihood estimate for p_1?

3. [**2 pt**] Now suppose that someone else observes the coin flip (still denoted by X) and tells you Y, the outcome of the flip, but this person only reports the correct result with probability p_2. Suppose we have the following dataset: X-the sequence of actual coin toss outcomes- is:

   $$(1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)$$

   Y- the sequence of coin toss outcomes we were told by the other person-is:

   $$(1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1)$$

   What is the maximum likelihood estimate for p_2?

4. [**2 pt**] Another person is observing the coin toss too, but the probability for this person to report the correct result depends on the actual outcome of the coin toss. Let $p_{a,b}$ be the probability for that person to report outcome b given that the actual outcome of the coin toss is a , where $a, b \in \{0, 1\}$. Consider the same X and Y values as given in previous question, what is the maximum likelihood estimate for $p_{0,0}$?

5. [**4 pt**] Let θ be a random variable with the probability density function:

   $$f(\theta) = \begin{cases} 2\theta, & \text{if } 0 \le \theta \le 1 \\ 0, & \text{otherwise.} \end{cases}$$

   Suppose that another random variable Y, conditioning on θ, follows an exponential distribution with $\lambda = 3\theta$. Note that the exponential distribution with parameter λ has a probability density function

   $$f(y) = \begin{cases} \lambda e^{-\lambda y}, & \text{if } y \ge 0, \\ 0, & \text{otherwise.} \end{cases}$$

Find the MAP estimate of θ given $Y = 4$ is observed.

6. **[1 pt] True or False**: If we choose an incorrect set of parameters for the beta prior of Bernoulli distribution, then the MAP estimate will not converge (as the number of training examples grows toward infinity) to the true value. (here, when we say an 'incorrect' set of parameters for the beta prior, we mean a set of parameters for which the most probable value is different from the true value of the parameter we are trying to estimate.)

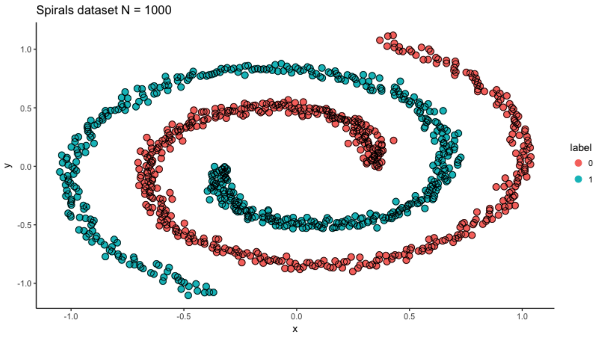
7. **[1 pt] True or False**: In case we choose Beta parameters that correspond to a uniform prior, the value of the MAP estimate will be identical to that of the MLE.

8. [**4 pt**] The next two questions refer to the following scenario: suppose that 0.5% people have cancer. Someone decided to take a medical test for cancer. The outcome of the test can either be positive (cancer) or negative (no cancer). The test is not perfect - among people who have cancer, the test comes back positive 96% of the time. Among people who don't have cancer, the test comes back positive 2% of the time. For the following questions, you should assume that the test results are independent of each other, given the true state (*cancer* or *no cancer*).

   What is the probability of a test subject having cancer, given that the subject's test result is positive?

9. [**4 pt**] In the same scenario as the previous question, a test subject's first test returned positive, and the subject decided to do a second independent test. The second test returned negative. What is the probability that this subject has cancer?

10. **[1 pt] True or False**: Gaussian Naive Bayes can be used to perfectly classify the training data shown below.



11. **[1 pt] Note**: This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    How many parameters will the model need under the Naïve Bayes assumption, assuming that $P(X_w = x_w|Y = y)$ is a Bernoulli distribution for each w and $P(Y = y)$ is also a Bernoulli distribution? All answers are shown as a function of the vocabulary size V.

    A. V

    B. $2V$

    C. $V + 1$

    D. $2V + 1$

12. **[1 pt] Note**: This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    How many parameters (also as a function of V) will the model need if we **do not** make the NB assumption, assuming $P(Y = y)$ is Bernoulli again and all of the features in X have binary labels?

    A. $2V$

    B. 2^V

    C. $2(2^V - 1) + 1$

    D. 2^{2V+1}

13. **[1 pt] Note**: This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    Does the Naïve Bayes assumption hold true for our dataset? Select a valid explanation for your answer.

    A. True. The appearances of each pair of words are not related regardless of review class.

    B. False. The appearances of some common stopwords (say, pronoun *he* and *she*) are dependent in both classes of movie reviews.

    C. True. The number of occurrences for words are not conditionally independent, but the appearances certainly do.

    D. False. For example, *Darth* and *Vader* are unlikely to be independent in both positive and negative reviews.

14. **[1 pt] Note: only one of the answers is correct.** This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    Which of the following statement(s) is/are correct with respect to using stopwords as features?

    A. We can keep stopwords as features. They have no effect on the accuracy of classifier.

    B. Stopwords add value to the dataset which is useful for correctly classifying the document.

    C. Removing stopwords helps in reducing noise/false positives.

    D. All of the above.

    E. None of the above.

15. **[1 pt] Note: only one of the answers is correct.** this is a single choice question. This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    We will experiment with two different parameter settings for our prior over θ_{yw}:

    (a) $\beta_0 = 5$ and $\beta_1 = 7$, and

    (b) $\beta_0 = 7$ and $\beta_1 = 5$.

    Train your classifier with 2 sets of data (`XTrainSmall`,`yTrainSmall`) and (`XTrain`,`yTrain`) with the first parameter setting. Then, use the learned classifiers to classify whether the reviews `XTest` are positive or negative. How do the classification errors compare?

    A. Error is smaller when using `XTrain`,`yTrain`.

    B. Error is smaller when using `XTrainSmall`,`yTrainSmall`.

    C. Errors are equal.

16. [**4 pt**] **Note**: This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    Train your classifier on the data contained in `XTrain` and `yTrain` with the second parameter setting in the previous problem. Then, use the learned classifier to classify whether the reviews `XTest` are positive or negative. After comparing classification errors produced by classifiers trained by `XTrain` and `yTrain` with 2 parameter settings, which parameter setting was a better choice for the prior on θ_{yw}?

    A. $\beta_0 = 5$ and $\beta_1 = 7$

    B. $\beta_0 = 7$ and $\beta_1 = 5$

17. [**4 pt**] **Note**: This question is based on material discussed in Part 2 - Implementing Naïve Bayes of the homework assignment. Please complete Part 2 of this assignment before attempting these questions.

    Consider again the Naïve Bayes classifiers trained with `XTrain` and `yTrain` for both parameter settings. Which of the settings of β_0 and β_1 make more sense if we strongly believe the true value of θ_{yw} lies in the interval $[0.1, 0.3]$?

    A. $\beta_0 = 5$ and $\beta_1 = 7$

    B. $\beta_0 = 7$ and $\beta_1 = 5$

18. [**0.5 pt**] **Collaboration Policy Question**: Did you receive any help whatsoever from anyone in solving this assignment? Please answer *yes* or *no*.

19. [**0.5 pt**] **Collaboration Policy Question**: If you answered *yes* on the previous question, please give full details below (e.g., *Christopher Nolan* explained to me what is asked in Question 3.4).

20. [**0.5 pt**] **Collaboration Policy Question**: Did you give any help whatsoever to anyone in solving this assignment? Please answer *yes* or *no*.

21. [**0.5 pt**] **Collaboration Policy Question**: If you answered *yes* on the previous question, please give full details below (e.g., I pointed *Michael Bay* to section 2.3 since he didn't know how to proceed with Question 2).

22. [**0.5 pt**] **Collaboration Policy Question**: Did you find or come across code that implements any part of this assignment? Please answer *yes* or *no*.

23. [**0.5 pt**] **Collaboration Policy Question**: If you answered *yes* on the previous question, please give full details below (book & page, URL & location, movies & scene, etc).

Part 2: Implementing Naïve Bayes [62 points]

For this part of the assignment, you must submit your code on Autolab. Once you have completed and submitted your code for Part 2, please return to "Part 1: Multiple Choice and Short answer questions" and complete the questions pertaining to "Part 2: Implementing Naive Bayes"

For this question, you will implement a Naïve Bayes (NB) classifier. You are given a dataset containing a training matrix generated from passages of movie reviews (in fact, the dataset you will be working on is a subset of IMDB Large Movie Review dataset used by Mass et. al, ACL 2011). Your task is to estimate appropriate parameters using the training data, and classify each passage as either a positive or a negative review.

The features used to classify passages are generated from the words (with stop words excluded) appearing in those reviews. A stop word is a word that is extremely common in literature which would appear to be of little value in helping a data scientist like you to build statistical and machine learning models for text mining. For example, whether the pronoun *I* is present probably won't add much useful information in telling if a movie review is positive. The set of all included words from all the passages in our dataset is called the *vocabulary*, and let's say its size is V. We will represent each article as a feature vector $\mathbf{x} = (x_1, ..., x_v)$, such that

$$x_j = \begin{cases} 1 & \text{if the } j^{\text{th}} \text{ word is present in the passage} \\ 0 & \text{otherwise.} \end{cases}$$

We also associate each review with a label y such that

$$y = \begin{cases} 0 & \text{if the review is negative} \\ 1 & \text{if the review is positive} \end{cases}$$

We make two key assumptions with the Naïve Bayes classifier. First, we assume our data are drawn *i.i.d* (independent and identically distributed) from a joint probability distribution over feature vectors $\mathbf{X}$ and labels Y. More importantly, we assume that for each pair of features X_i and X_j with $i \neq j$, X_i is conditionally independent of X_j given the class label Y (alas, how *naïve* it is to assume the appearances of *Harry* and *Potter* is unrelated in a positive movie review). To predict the label of an article, we choose the most probable class label given $\mathbf{X}$.

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y|\mathbf{X} = \mathbf{x})$$

Using the Bayes rule and the NB assumption, we can rewrite the above expression as follows

$$\begin{aligned}
\hat{y} &= \underset{y}{\operatorname{argmax}}\, \frac{P(\mathbf{X} = \mathbf{x}|Y = y)P(Y = y)}{P(\mathbf{x})} && \text{(Bayes' rule)} \\
&= \underset{y}{\operatorname{argmax}}\, P(\mathbf{X} = \mathbf{x}|Y = y)P(Y = y) && \text{(denominator does not depend on } y\text{)} \\
&= \underset{y}{\operatorname{argmax}}\, P(X_1 = x_1, \ldots, X_V = x_V|Y = y)P(Y = y) \\
&= \underset{y}{\operatorname{argmax}} \prod_{w=1}^{V} P(X_w = x_w|Y = y)P(Y = y) && \text{(Naïve Bayes assumption)}
\end{aligned}$$

By making the NB assumption, we can factor the probability distribution $P(\mathbf{X} = \mathbf{x}|Y = y)$ as a product of all $P(X_w = x_w|Y = y)$, allowing us to define and learn significantly fewer parameters.

1. **[1 point]** How many parameters will the model need under the Naïve Bayes assumption, assuming that $P(X_w = x_w|Y = y)$ is a Bernoulli distribution for each w and $P(Y = y)$ is also a Bernoulli distribution? All answers are shown as a function of the vocabulary size V. Please submit your answer in Part 1 writeup.

2. **[1 point]** How many parameters (also as a function of V) will the model need if we **do not** make the NB assumption, assuming $P(Y = y)$ is Bernoulli again and all of the features in X have binary labels? Please submit your answer in Part 1 writeup.

3. **[2 points]** Does the Naïve Bayes assumption hold true for our dataset? Select a valid explanation for your answer. Please submit your answer in Part 1 writeup.

4. **[1 point]** Which of the following statement(s) is/are correct with respect to using the appearances of stopwords as features? Please submit your answer in Part 1 writeup.

Since we do not know the true joint distribution over $\mathbf{X}$ and Y, we need to estimate $P(\mathbf{X} = \mathbf{x}|Y = y)$ and $P(Y = y)$ from the training data. For each word w and class label y, suppose that the distribution of X_w given Y is a Bernoulli distribution with the parameter θ_{yw}, such that

$$P(X_w = 1|Y = y) = \theta_{yw} \quad \text{and} \quad P(X_w = 0|Y = y) = 1 - \theta_{yw}$$

A common problem in language related ML problems is dealing with words not seen in training data. Without any prior information, the probability of unseen words is zero. We know that this is not a good estimate, and we want to assign a small probability to any word in the vocabulary occurring in either positive or negative reviews. We can achieve that by imposing a Beta(β_0, β_1) prior on θ_{yw}, and perform a MAP estimate from the training data. The p.d.f. of the Beta distribution is given as follows:

$$f(\theta; \beta_0, \beta_1) = \frac{1}{\mathbf{B}(\beta_0, \beta_1)} \theta^{\beta_0 - 1}(1 - \theta)^{\beta_1 - 1} \ \forall y, w, \theta$$

where $\mathbf{B}(\beta_0, \beta_1)$ is the beta function:

$$\mathbf{B}(\beta_0, \beta_1) = \int_0^1 t^{\beta_0 - 1}(1 - t)^{\beta_1 - 1} dt$$

You will experiment with two combinations of β_0 and β_1 values in this problem. Assume the distribution of Y is a Bernoulli distribution (taking values 0 or 1), as given below.

$$P(Y = 0) = \phi \quad \text{and} \quad P(Y = 1) = 1 - \phi$$

Since we have enough reviews in both classes, we need not worry about zero probabilities, and will not impose a prior on ϕ.

Programming Instructions

You will implement some functions for training and testing a Naïve Bayes classifier for this question. You will submit your code online through the CMU autolab system, which will execute it remotely against a suite of tests. Your grade will be automatically determined from the testing results. **Reminder: you can submit your code up to 30 times on Autolab**.

To get started, you can log into the autolab website (`https://autolab.andrew.cmu.edu`). From there you should see 10-601 in your list of courses. Download the handout for Homework 3 (Options → Download handout) and extract the contents (i.e., by executing `tar xvf hw3.tar` at the command line). In the archive you will find three folders. The `data` folder contains the data files for this problem. The `python` folder contains a `NB.py` file which contains empty function templates for each of the functions you are asked to implement. Similarly, the `octave` folder contains separate `.m` files for each of the functions that you are asked to implement.

To finish each programming part of this problem, open the `NB.py` or the function-specific `.m` template files and complete the function(s) defined inside. When you are ready to submit your solutions, you will create a new tar archive of the files you are submitting the `hw3` directory. Please create the tar archive exactly as detailed below.

If you are submitting Python code:

```
tar cvf hw3_handin.tar NB.py
```

If you are submitting Octave code:

```
tar cvf hw3_handin.tar logProd.m NB_XGivenY.m NB_YPrior.m NB_Classify.m classificationError.m
```

If you are working in Octave and are missing **any** of the function-specific `.m` files in your tar archive, you will receive zero points.

We have provided all of the data for this assignment as `csv` files in the `data` folder in your handout. You can load the data using the `numpy.genfromtext` function in Python or the `csvread` and `csv2cell` (`io` package) functions in Octave. We have provided a `hw3_script` file for each language to help get you started and load data into your work space (note for `Octave` users: loading the dataset may take a little extra time depending on your computer specs). You should build on the `hw3_script` to test out the functions we are asking you to implement and turn in, but you will not submit the script itself. After loading the data, you should have 7 variables: `vocabulary`, `stopword`, `XTrain`, `yTrain`, `XTest`, `yTest`, `XTrainSmall`, and `yTrainSmall`.

- `vocabulary` is a $V \times 1$ dimensional vector that contains every word, excluding stop words, appearing in the passages. When we refer to the j^{th} word, we mean `vocabulary(j)`. You will not need the vocabulary vector for any of the questions on this homework, but we have included it to help you better understand the data set.
- `stop_words` is a $S \times 1$ dimensional vector that contains every stop word appearing in the passages. Also included for completeness purposes.
- `XTrain` is a $n \times V$ dimensional matrix describing the n documents used for training your Naïve Bayes classifier. The entry `XTrain(i,j)` is 1 if word j appears in the i^{th} training passage and 0 otherwise.
- `yTrain` is a $n \times 1$ dimensional vector containing the class labels for the training documents. `yTrain(i)` is 0 if the i^{th} passage is a negative review and 1 if it is a positive one.
- `XTest` and `yTest` are the same as `XTrain` and `yTrain`, except instead of having n rows, they have m rows. This is the data you will test your classifier on and it should not be used for training.
- Finally, `XTrainSmall` and `yTrainSmall` are subsets of `XTrain` and `yTrain` which are used in the final question.

Code

Preliminary Note

For this programming assignment, you are not allowed to use any machine learning packages in `python` or `octave` (e.g., `scikit-learn`). For `python` users, you are restricted to use only the `math`, `numpy`, and `scipy` libraries. For `octave` users, you are not allowed to use any external libararies.

Preliminary Note for Python Users

In `numpy`, to initialize an $n \times 1$ dimensional vector, you should assign `y = numpy.array([y1,y2,...,yn])`, and to check its dimension, we use `y.shape` which yields `(n,)`. You **should not** initialize a vector by means of initializing a matrix, that is `X = np.ones((n,1))`, which will output a matrix with dimension `X.shape = (n,1)`. To initialize a $m \times n$ matrix with all entries equal to 1, we use `np.ones((m,n))`.

Logspace Arithmetic

When working with very large or very small numbers (such as probabilities), it is useful to work in *logspace* to avoid numerical precision issues. In logspace, we keep track of the logs of numbers, instead of the original values. For example, if $p(x)$ and $p(y)$ are probability values, instead of storing $p(x)$ and $p(y)$ and computing $p(x) * p(y)$, we work in log space by storing $\log(p(x))$, $\log(p(y))$, and we can compute the log of the product, $\log(p(x) \times p(y))$ by taking the sum: $\log(p(x) \times p(y)) = \log(p(x)) + \log(p(y))$.

5. [**8 points**] Complete the function `logProd(x)` which takes an input a vector of numbers in logspace (i.e., $x_i = \log p_i$) and returns the product of those numbers in logspace—i.e., `logProd(x)` $= \log\big(\prod_i p_i\big)$.

Training Naïve Bayes

6. [**12 points**] Complete the function `NB_XGivenY(XTrain, yTrain, beta_0, beta_1)`. The output `D` is a $2 \times V$ matrix, where for any word w and class label y, the entry `D(y,w)` is the MAP estimate of $\theta_{yw} = P(X_w = 1|Y = y)$ with a Beta(β_0, β_1) prior distribution. A useful initial step will be to derive the closed form expression for the MAP estimate of Bernoulli parameter θ_{yw} with a Beta prior. Note that the parameters of the Beta distribution are also given as input to the function.

7. [**12 points**] Complete the function `NB_YPrior(yTrain)`. The output `p` is the MLE for $\phi = P(Y \ = \ 0)$.

8. [**20 points**] Complete the function `NB_Classify(D, p, XTest)`. The input `XTest` is an $m \times V$ matrix containing m feature vectors (stored as its rows). D and p are in the same form of outputs of question 6 and 7. The output `yHat` is a $m \times 1$ vector of predicted class labels, where `yHat(i)` is the predicted label for the i^{th} row of `XTest`. [Hint: In this function, you will want to use the `logProd` function to avoid numerical problems.]

9. [**10 points**] Complete the function `classificationError(yHat, yTruth)`, which takes two vectors of equal length and returns the fraction of entries that disagree.

Experiments

10. [**1 points**] We will experiment with two different parameter settings for our prior over θ_{yw}:

    (a) $\beta_0 = 5$ and $\beta_1 = 7$, and

    (b) $\beta_0 = 7$ and $\beta_1 = 5$.

    Train your classifier with 2 sets of data (`XTrainSmall`,`yTrainSmall`) and (`XTrain`,`yTrain`) with the first parameter setting. Then, use the learned classifiers to classify whether the reviews `XTest` are positive or negative. How do the classification errors compare? Please submit your answer in Part 1 writeup.

11. [**4 points**] Train your classifier on the data contained in `XTrain` and `yTrain` with the second parameter setting in the previous problem. Then, use the learned classifier to classify whether the reviews `XTest` are positive or negative. After comparing classification errors produced by classifiers trained by `XTrain` and `yTrain` with 2 parameter settings, which parameter setting was a better choice for the prior on θ_{yw}? Please submit your answer in Part 1 writeup.

12. [**4 points**] Consider again the Naïve Bayes classifiers trained with `XTrain` and `yTrain` for both parameter settings. Which of the settings of β_0 and β_1 make more sense if we strongly believe the true

value of θ_{yw} lies in the interval $[0.1, 0.3]$? Please submit your answer in Part 1 writeup. You do not need to submit your plots.