Machine Learning: Homework 3

Skye Chen,zc969@nyu.edu

Due 11:55 p.m. Friday, March 14, 2020

Instructions

- **Collaboration policy:** Homeworks must be done individually, except where otherwise noted in the assignments. "Individually" means each student must hand in their own answers, and each student must write and use their own code in the programming parts of the assignment. It is acceptable for students to collaborate in figuring out answers and to help each other solve the problems, though you must in the end write up your own solutions individually, and you must list the names of students you discussed this with. We will be assuming that, as participants in an undergraduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.
- **Format of Submission:** Please submit your homework with a single zip file. Files in the zip file are
  - **NetID-HW3.pdf** : a write-up file which contains answers to problems, and **it should contain the commands for running your code**;
  - **problem-X.py** : the code files for problems (where X is the ID of the problem).
- **Online submission:** You must submit your solutions online on NYU Classes. The write-up file should be in PDF format. We recommend that you use LaTeX. PDF files exported from doc/docx files are also accepted. Please do not submit hand-written solutions.
- **skeleton code:** We provide skeleton code files for Problem 1 and Problem 3. Please use them.

Instruction for the code

The zip file contains 3 .py file each labeled with problem number. To run the code, you just run the file and type training data file in terminal. If you want to plot figures, you should un-comment the commented part. It should be very straightforward.

Problem 1: Naïve Bayes

Using the same spam email dataset as we used in Homework 1, implement a Naïve Bayes classifier for spam email classification. See **naive-bayes-skeleton.py** for the skeleton code.

(a) [**5 Points**] Use boolean features as we did in Homework 1, i.e., x_j being 1 if the j^{th} word in the vocabulary occurs in the email, or 0 otherwise.
Solution:
See code Problem1.py. After compare MLE and MAP, MAP gives a smaller validation error(5.2%), while MLE is 5.7%. Therefore is used in the following problems.

(b) [**5 Points**] Plot the training and validation errors as a function of training size N. (Hint: This is the same as what you did in Homework 1, Problem 1.5. Also, you may use log-scale to avoid numerical issues.)

Solution:
The plot is 1 for using MAP. The code for plotting is commented in the code. The value of N tried is: 200,600,1200,2400,4000, which is the same as HW 1. The plot shows us that $N = 4000$ gives the smallest validation error rate, therefore this value is used for the following questions.

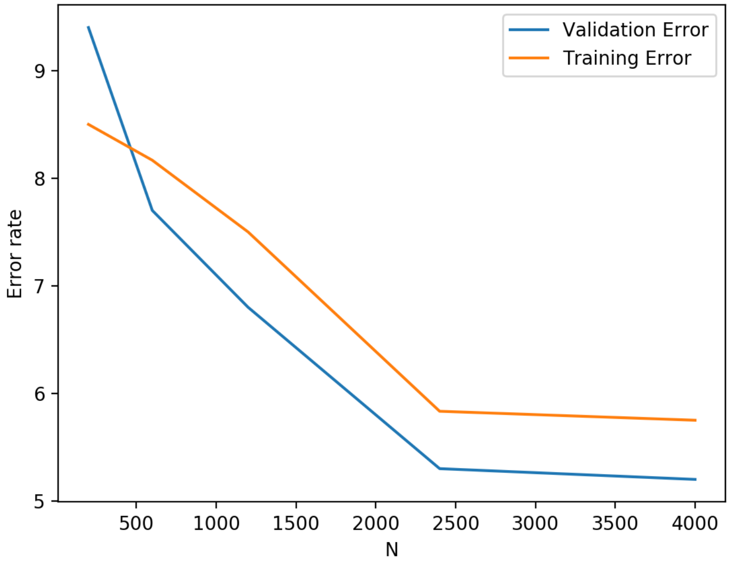

Figure 1: Validation and training error verses size N with MAP

(c) [**5 Points**] To improve the accuracy, feel free to change the dimensionality of features by using a different vocabulary threshold X. Tell me about what you try and what result you get.
Solution:
Try X = 22:
Validation error, # = 54, % = 5.4000%. Training error, # = 230, % = 5.7500%.
Try X = 26:
Validation error, # = 52, % = 5.2000%. Training error, # = 230, % = 5.7500%.
Try X = 28:
Validation error, # = 52, % = 5.2000%. Training error, # = 228, % = 5.7000%.
Try X = 30:
Validation error, # = 51, % = 5.1000%. Training error, # = 222, % = 5.5500%.
Therefore, we choose X = 30 for following questions.

(d) [**5 Points**] Try both MLE and MAP estimators (For simplicity, just set the hallucinated word count to be 1). Which one is better? Please briefly explain. (For the previous two questions, you only have to report the result of either MLE or MAP, whichever is better according to your experiment.)
Solution:
From previous questions, the parameter value used for test is $N = 4000$, $X = 30$.
For MAP, Test error, # = 65, % = 6.5000%. The comparison is made while answering problem 1a. MAP gives a better result since hallucinated term is added to avoid probability being 0 and it is more accurate.

Problem 2 (Bonus): Naïve Bayes with Bag of Words

[**+10 Points**] Repeat Problem 1 using the “bag of words” features with Naïve Bayes classifier.

(a) **[5 Points]** Use boolean features as we did in Homework 1, i.e., x_j being 1 if the j^{th} word in the vocabulary occurs in the email, or 0 otherwise.
See Code Problem2.py
Clarification: My whole thought of this problem is one we discussed in Thursday's office hour, i.e. feature vector is the times of a word shows up.
For MLE:
Validation error, # = 39, % = 3.9000%. Training error, # = 89, % = 2.2250%.
For MAP:
Validation error, # = 27, % = 2.7000%. Training error, # = 104, % = 2.6000%. MAP is better since it prevents 0 values by adding hallucinated term.

(b) **[5 Points]** Plot the training and validation errors as a function of training size N. (Hint: This is the same as what you did in Homework 1, Problem 1.5. Also, you may use log-scale to avoid numerical issues.)
Solution:
See Figure 2. From figure, we can see that N = 4000 gives the smallest validation error, therefore is used in following questions.

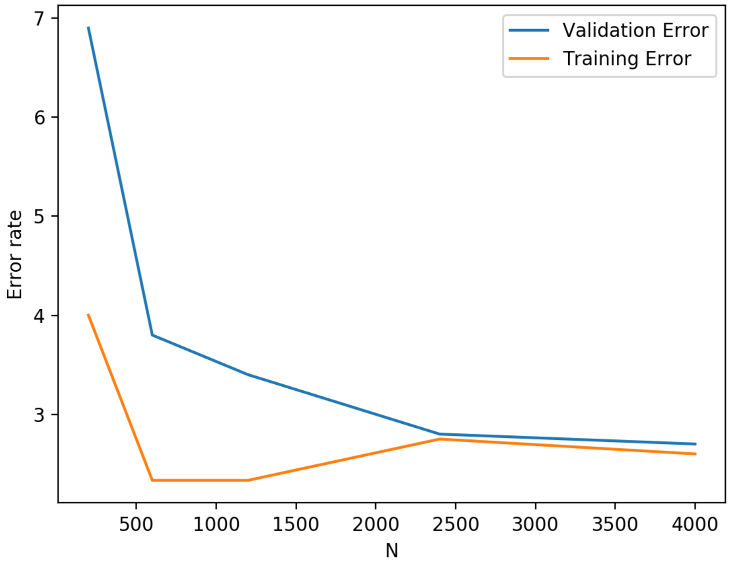

Figure 2: Validation and training error verses size N with MAP

(c) **[5 Points]** To improve the accuracy, feel free to change the dimensionality of features by using a different vocabulary threshold X. Tell me about what you try and what result you get.
Solution:
Try X = 22:
Validation error, # = 27,% = 2.7000%. Training error, # = 101, % = 2.5250%.
Try X = 26:
Validation error, # = 27, % = 2.7000%. Training error, # = 101, % = 2.6000%.
Try X = 28:
Validation error, # = 27, % = 2.7000%. Training error, # = 103, % = 2.5750%.
Try X = 30:
Validation error, # = 26, % = 2.6000%. Training error, # = 107, % = 2.6750%.
Therefore, we should use X = 30 for the following question.

(d) [**5 Points**] Try both MLE and MAP estimators (For simplicity, just set the hallucinated word count to be 1). Which one is better? Please briefly explain. (For the previous two questions, you only have to report the result of either MLE or MAP, whichever is better according to your experiment.)
Solution:
Test error, # = 33, % = 3.3000%.
For this question, if we do not use MAP, we will have invalid value error and it is very inaccurate. Therefore, the MAP should be employed.

Problem 3: Gaussian Naïve Bayes with Bag of Words

[**20 Points**] Repeat Problem 1 using the "bag of words" features with **Gaussian** Naïve Bayes classifier. See **gaussian-nb-skeleton.py** for the skeleton code.

(a) [**5 Points**] Use boolean features as we did in Homework 1, i.e., x_j being 1 if the j^{th} word in the vocabulary occurs in the email, or 0 otherwise.
See Code Problem3.py
Clarification: In this problem, only MAP is used since if we do not add hallucinated term, there will be some case where the standard deviation is 0. The way for my MAP is that I added two hallucinated email(of spam and non-spam) containing every word in the wordlist. In prediction, I took the log-form of likelihood function and simplified the equation of the pdf of normal distribution. The scenario I used in this problem is the same as the one in Problem 2.

(b) [**5 Points**] Plot the training and validation errors as a function of training size N. (Hint: This is the same as what you did in Homework 1, Problem 1.5. Also, you may use log-scale to avoid numerical issues.)
Solution:
See Figure 3. From figure, we can see that N = 600 gives the smallest validation error, therefore is used in following questions.
N = 600, Validation error, # = 88,% = 8.8000%.

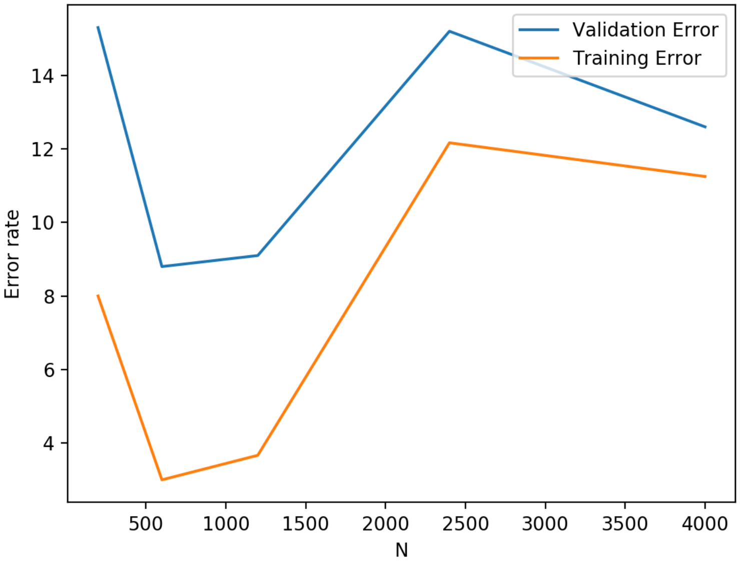

Figure 3: Validation and training error verses size N with MAP

(c) **[5 Points]** To improve the accuracy, feel free to change the dimensionality of features by using a different vocabulary threshold X. Tell me about what you try and what result you get.
Solution:
Try X = 22:
Validation error, # = 84, % = 8.4000%. Training error, # = 21, % = 3.5000%.
Try X = 26:
Validation error, # = 88, % = 8.8000%. Training error, # = 18, % = 3.0000%.
Try X = 28:
Validation error, # = 82, % = 8.2000%. Training error, # = 19, % = 3.1667%.
Try X = 30:
Validation error, # = 85, % = 8.5000%. Training error,# = 20, % = 3.3333%.
From above data, we know that X = 28 gives the smallest error rate, therefore it is used in following question.

(d) **[5 Points]** Try both MLE and MAP estimators (For simplicity, just set the hallucinated word count to be 1). Which one is better? Please briefly explain. (For the previous two questions, you only have to report the result of either MLE or MAP, whichever is better according to your experiment.)
Solution:
Test error, # = 130, % = 13.0000%.
Since we can not use MLE in this question, there is no need to make comparison.

Problem 4: Comparison

[5 Points] Now you have mastered three algorithms to do spam email classification: Perceptron, Naïve Bayes, and Gaussian Naïve Bayes. Briefly summarize their pros and cons based on your understanding and experiment. (Keep it less than 150 words.)
Solution:
Naïve Bayes is much faster than Perceptron, therefore gives a high speed. However, it depends on the assumption of conditional independence, which is not really valid in real cases, therefore is not that accurate.
Gaussian Naive Bayes assumes the Gaussian distribution of data, however, will be problematic when the standard deviation is very small, for example, in Problem 3. From the result, we can easily find out that the error is much larger than previous ones. It also gives problem since x is integer value whereas the mean is usually smaller than 1. It is also fast, though.
Perceptron is easy to implement, however costs longer time to train and highly requires that the data is line-separable.