CS6923 Machine Learning, Spring 2019
Prof. Sellie, NYU School of Engineering

Homework 1

Submit on NYU Classes by Mon. Feb. 13 at 8:00 p.m. Submit two files: (1) a **pdf** file with your written answers for Part I. (2) a **zip** file including code and answers for Part II. You do not have to typeset your written answers: as long as your handwriting is readable, you can write out the answers by hand and scan your answers. If you do that, use a utility like camscanner to make sure that the scan is clear.

You may work together with one other person on this homework. If you do that, *hand in JUST ONE homework for the two of you*, with both of your names on it. You may *discuss* this homework with other students but YOU MAY NOT SHARE WRITTEN ANSWERS OR CODE WITH ANYONE BUT YOUR PARTNER.

Part I: Written Exercises

1. You are a geologist. You collect rocks from two locations, Location A and Location B.

   The rocks from Location A tend to be heavier than those from Location B. The rocks from Location A are distributed according to a Gaussian distribution, with mean $\mu_1 = 9.2$ grams and standard deviation $\sigma_1 = 1.6$ grams. The rocks from Location B are distributed according to a Gaussian distribution, with mean $\mu_2 = 9.6$ grams and standard deviation $\sigma_2 = 1.2$ grams.

   In your laboratory, you find 3 rocks that you collected at the same location. Unfortunately, you forgot to label them, and don't know whether you collected them at Location A or Location B.

   (a) Suppose you believe that the probability that the rocks are from Location B is four times the probability that the rocks are from Location A.

   According to this belief, what is the probability that it was from Location A? What is the probability that it was from Location B? (Make sure these 2 probabilities sum to 1.)

   (b) Now suppose you weigh the 3 rocks, and they weigh $9.3, 8.8$ and 9.8 grams. Using the probabilities from part (a) as your priors, and the weights of the 3 rocks as your data (evidence), compute the *posterior probability* that the rocks are from Location A.

   (c) Ignoring the priors, and just focusing on the weights, which is the ML hypothesis?

2. Suppose there are two types of screenings that are used to detect whether a person has a certain disease. Screening method 1 is a blood test which is fast and inexpensive, but not very accurate. It has a 15% false positive rate, and a 10% false negative rate.

   Screening method 2 is based on doing an MRI of the patient. It is expensive and takes more time. It has a 5% false positive rate, and a 3% false negative rate.

   The "false positive rate" is the probability that the method mistakenly returns a positive result (says the person has the disease), when the person does not have the disease. The "false negative rate" is the probability that the method mistakenly returns a negative result (says the person doesn't have the disease), when the person *does* have the disease.

   Suppose that 0.02% of the population has this disease.

   (a) Suppose a random person from the population is screened using Screening method 1, and it returns a positive result. Using a Bayesian approach, which is the MAP hypothesis: that the person has the disease, or that the person does not have the disease?

   (b) Repeat the previous question, but give the ML hypothesis.

   (c) Now suppose that because the person had a positive result using Screening method 1, the person is then sent to be screened using Screening method 2. If the result is positive again, what is the posterior probability that the person has the disease? Assume the results of the two screening methods are independent.

3. Consider a coin with unknown bias θ, where θ is the probability of Heads. Suppose you flip that coin 6 times and get TTHTTH. (T is for tails, and H is for heads.)

   (a) As a function of θ, what is $P(TTHTTH|\theta)$?

   (b) As a function of θ, what is $\log P(TTHTTH|\theta)$?

   Express your answer as a function of θ, in the form $a \log \theta + b \log(1-\theta)$, for some constants a and b.

   (c) Find the *maximum likelihood estimate* of θ. That is, find $argmax_\theta P(TTHTTH|\theta)$.

   Because computing $argmax_\theta P(HTTHH|\theta)$ is messy, use the log trick, and instead compute $argmax_\theta \log P(HTTHH|\theta)$.

   You will probably need to use calculus to answer this question.

4. Let $X_1, \ldots, X_N$ be discrete i.i.d. random variables, taking on values from the set $\{1, \ldots, s\}$ for some integer s.

   For $v \in \{1, \ldots, s\}$, let θ_v denote $P[X_i = v]$.

   You can think of the X_i's as being the outcomes of N rolls of a die with s sides, where the probability of side v is θ_v.

Consider one particular v. Suppose we do not know θ_v, but want to estimate it using the sample $\mathcal{X} = \{X_1, \dots, X_n\}$. Let t denote the number of occurences of v in $\mathcal{X}$.

Consider the following estimates of θ_v for $\mathcal{X}$:

(i) $\frac{t}{N}$ (an unsmoothed estimate)

(ii) $\frac{t+1}{N+s}$ (a smoothed estimated using add-1 smoothing)

(iii) $\frac{t+m}{N+sm}$ for some fixed $m > 0$ (a smoothed estimate using add-m smoothing)

For example, suppose $s = 2$ and $v = 1$. If $\mathcal{X} = \{1, 2, 1, 1\}$, then the above estimates for θ_1 given $\mathcal{X}$ are $\frac{3}{4}, \frac{4}{6}, \frac{3+m}{4+2m}$.

The above estimates are related. If we inserted s additional X_i into $\mathcal{X}$, one for each possible value in the set $\{1, 2, ..., s\}$, we would have a sample of size $N+s$ with $t+1$ v's; calculating the unsmoothed estimate for this expanded sample yields estimate (ii). Similarly, suppose we inserted ms additional X_i into $\mathcal{X}$, with m copies for each value in the set $\{1, 2, ..., s\}$; calculating the unsmoothed estimate for this expanded sample yields estimate (c). If you set $m = 1$ in Estimate (iii), you get Estimate (ii) (add-1 smoothing).

Estimate (i) is the simplest, as it is just the fraction of v's in the sample. But there are reasons to use a smoothed estimate instead. One is to avoid having the estimate be equal to 0. In practice many people use add-1 smoothing, but this is generally a bad idea. It is often better to use estimate (iii) with a value of m that is between 0 and 1 (e.g., $m = 0.01$).

(a) Estimate (i) is the ML estimate of θ, given the sample $\mathcal{X}$. This can be proved using reasoning similar to that used in Problem 3, but you do not have to prove it here. Instead, answer the following simple question:

If $s = 3$, what is the ML estimate for θ_1, given $\mathcal{X} = \{3, 1, 1, 2, 3\}$?

(b) Another simple question: If $s = 3$, what is the estimate for θ_1 using add-1 smoothing, given $\mathcal{X} = \{3, 1, 1, 2, 3\}$?

(c) Suppose $s = 2$.

Consider the prior distribution on θ_1 that has pdf $p(\theta_1) = 6\theta_1(1-\theta_1)$ when $\theta_1 \in [0, 1]$, and $p(\theta_1) = 0$ otherwise.. (This is a valid pdf, because $\int_0^1 6\theta_1(1 - \theta_1) = 1$.)

What is the MAP estimate of θ_1, for a generic sample $\mathcal{X}$ of size N with t 1's? Express your answer as a function of t and N. Show your work. Hint: Your answer should be the same one that you would get by using add-m smoothing, for some value of m.

5. In this problem, you will use the Naive Bayes algorithm on a simple dataset, to predict the label of a new example. We did not finish covering the Naive Bayes algorithm in class on Thursday (Sept. 13th), but the following questions will teach you what you need to know.

Consider the following small labeled dataset for a binary classification problem (classes + and -). There are three categorical attributes, x_1, x_2, and x_3. (A categorical attribute takes on values from a small, unordered set.) Attribute x_1 takes on values from the set $\{Medium, Low, High\}$, x_2 takes on values from the set $\{No, Yes\}$, and x_3 takes on values from the set $\{Red, Green\}$.

x1	x2	x3	label
Low	No	Red	+
Medium	No	Green	+
Low	Yes	Red	-
High	No	Green	-
Medium	Yes	Green	-

In the following questions, you will be asked to estimate quantities of the form $P(x_i = v|C)$ from the above dataset. You should do this by considering the subsample consisting only of the examples with label C, and applying the appropriate estimation method from the previous problem to this subsample.

For example, suppose you were asked to estimate $P(x_1 = Medium|+)$, using add-m smoothing, for $m = 0.3$. Since x_1 has 3 possible values, there are 2 examples labeled +, and only 1 has $x_1 = Medium$, the estimate would be $\frac{1+0.3}{2+0.3*3}$. Similarly, the estimate of $P(x_1 = High|+)$ would be $\frac{0+0.3}{2+0.3*3}$.

(a) Using add-m smoothing with $m = 0.3$, calculate estimates for $P(x_1 = Low|+)$, $P(x_2 = Yes|+)$, and $P(x_3 = Green|+)$. Then calculate estimates for $P(x_1 = Low|-)$, $P(x_2 = Yes|-)$, and $P(x_3 = Green|-)$.

(b) The Naive Bayes assumption says that the attributes are independent given the class.

Thus using the Naive Bayes assumption, if $x = [Low, Yes, Green]$, then

$$P(x|C) = P(x_1 = Low|C) * P(x_2 = Yes|C) * P(x_3 = Green|C)$$

Use this equation, and the estimates you calculated in part (a), to calculate (estimates of) $P(x|+)$ and $P(x|-)$, for $x = [Low, Yes, Green]$.

(c) Based on the estimates of $P(x|+)$ and $P(x|-)$ that you just calculated, which is the ML label for the example $x = [Low, Yes, Green]$?

(d) For the above dataset, an unsmoothed estimate of the priors $P(+)$ and $P(-)$, would be $P(+) = 2/5$ and $P(-) = 3/5$. Using these priors, and the values you calculated for $x = [Low, Yes, Green]$, what is the MAP label for the example $x = [Low, Yes, Green]$?

You have just used the Naive Bayes algorithm to classify a new example, given a training set.

Part II: Programming and Questions

Note: You must do your programming in Python.

Together with this homework, we have posted a training and test files for a simple spam classification problem. (Real-world spam classification tasks today are much more difficult!)

An example in this dataset corresponds to an email, and the attributes correspond to properties of the email. The problem is to classify the example into one of 2 classes: spam or not-spam. [1]

Files: The essential information about the data is given in the file `spambase1.txt`. There are 9 continuous input attributes, as described in the file. The class label is designated as 1 (spam) or 0 (non-spam).

The examples are given in two files:

the training examples are in `spambasetrain.csv`

the test examples are in `spambasetest.csv`.

Each line of these two files gives the information for one example. The first 9 columns have the values of the 9 input attributes, in the order specified in the `spambase1.txt` file. So, for example, column 3 contains the value for the attribute `char_freq_[`. The class of the example is given in the final column.

To do: Implement Gaussian Naive Bayes in Python.

Gaussian Naive Bayes is very similar to the standard Naive Bayes algorithm described in the written part of this homework (and in the Lecture 2 slides), but it is designed for datasets with numeric attributes. As in standard Naive Bayes, it uses the conditional independence assumption of Naive Bayes, which says that

$$p(x_1, x_2, \ldots, x_d|C) = p(x_1|C) * p(x_2|C) * \ldots * p(x_d|C)$$

It also assumes that the pdf $p(x_j|C)$ is a Gaussian pdf. It estimates this pdf by estimating the mean μ and the variance σ^2 of the distribution from the training examples.

For each test example $(x_1, \ldots, x_d)$, you will need to find the class maximizing

$$p(x_1|C) * p(x_2|C) * \ldots * p(x_d|C) * P(C)$$

using the estimates of the $p(x_j|C)$ and $p(C)$.

Training details: During the training phase, you will use the examples in the training file to estimate the value of $P(C)$ for each class. For each pair x_j, C

[1] The original dataset is taken from the UCIrvine data repository, and is available at `https://archive.ics.uci.edu/ml/datasets/spambase`. We have modified the dataset slightly for this assignment, so you should use the version that is posted on NYU Classes. Do not use the original version of the dataset for this assignment.

of attribute and class, you will need to estimate the mean and the Gaussian for the pdf $p(x_j|C)$.

Estimation of $P(C)$: To estimate the $P(C)$ values, just calculate the fraction of the training examples that are in class C. For example, to estimate $P(C=1)$ calculate:

$$\frac{\text{number of examples in training file in Class 1}}{\text{total number of examples in training file}}$$

Estimation of parameters of Gaussian pdf $p(x_i|C)$: Find all examples in the training file that are in class C. Let $\mathcal{X}_C$ denote this set. Let N_C be the number of examples in this set.

To estimate the mean $\hat{\mu}$ of the Gaussian pdf $p(x_i|C)$, just calculate the average value of attribute x_i among all examples in $\mathcal{X}_C$.

To estimate the variance, use the following formula:

$$\hat{\sigma}^2 = \frac{\sum_{(x^t,r^t)\in\mathcal{X}_C}(x_t - \hat{\mu})^2}{N_C - 1}$$

where $\hat{\mu}$ is the estimate of the mean that you just calculated.

Note that $N_C - 1$ is in the denominator here, not N_C. (We'll explain why later.)

Once these values are computed, the resulting estimated pdf is

$$p(x_i|C) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\hat{\mu})^2}{2\hat{\sigma}^2}} \tag{1}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are the values calculated for this x_j, C.

(Make sure to calculate a new $\hat{\mu}$ and $\hat{\sigma}^2$ for each x_j, C pair.)

Testing details: For each example $(x_1, \ldots, x_9)$ in the test file, you want to determine which class maximizes

$$p(x_1|C) * p(x_2|C) * \ldots * p(x_d|C) * P(C)$$

Use ONLY attributes 1 through 9, when doing testing. Do not accidentally include the label! Compute a value for each $p(x_j|C)$ using the pdf in Equation **??**, with the mean and variance you estimated for pair x_j, C during training.

Multiplying lots of small values can lead to underflow. To avoid that, you should not calculate

$$p(x_1|C) * p(x_2|C) * \ldots * p(x_9|C) * P(C)$$

directly for each class C. Instead, calculate $\log[p(x|C) * P(C)] = \log P(C) + \sum_{i=1}^{9} \log p(x_9|C)$. (Use the natural log, ln.) Then label the example with the class achieving the maximum value for this expression. If there is a tie, give the example the label 1.

Outputs

Your program should output the following values and write them to a separate text file (or the standard output) which you will NOT hand in.

- The estimated value of $P(C)$ for each class C.
- The estimates $(\hat{\mu}, \hat{\sigma}^2)$ for the Gaussians corresponding to $p(x_i|C)$, for each attribute x_i and each class C_i. (so you need to output 18 pairs $(\hat{\mu}, \widehat{\sigma^2})$).
- The predicted classes for all the test examples.
- Total number of test examples classified correctly. (You need to compare the predicted class for each test example with the given class label.)
- Total number of test examples classified incorrectly.
- The percentage error on the test examples.

Questions

Answers the following questions and put your answers in a pdf file called `proganswers.pdf`.

1. What was the estimated value of $P(C)$ for $C = 1$?
2. What was the estimated value of $P(C)$ for $C = 0$?
3. What were the estimated values for $(\hat{\mu}, \widehat{\sigma^2})$ for the Gaussian corresponding to attribute `capital_run_length_longest` and class 1 (Spam).
4. What were the estimated values for $(\hat{\mu}, \widehat{\sigma^2})$ for the Gaussian corresponding to attribute `char_freq_;` and Class 0.
5. Which classes were predicted for the first 5 examples in the test set?
6. Which classes were predicted for the last 5 examples in the test set?
7. What was the percentage error on the examples in the test file?
8. Sometimes a not-very-intelligent learning algorithm can achieve high accuracy on a particular learning task simply because the task is easy. To check for this, you can compare the performance of your algorithm to the performance of some very simple algorithms. One such algorithm just predicts the majority class (the class that is most frequent in the training set). This algorithm is sometimes called Zero-R. It can achieve high accuracy in a 2-class problem if the dataset is very imbalanced (i.e., if the fraction of examples in one class is much larger than the fraction of examples in the other). What accuracy is attained is you use Zero-R instead of Gaussian Naive Bayes?

9. Gaussian Naive Bayes is based on two assumptions: (1) the conditional independence assumption, and (2) the assumption that the pdfs for $p(x_j|C)$ are Gaussian. These assumptions are more reasonable for some datasets than for others. Do you think these assumptions are reasonable for the spam dataset you just used? Why or why not? In answering this question, you can give a common-sense argument and/or show relevant plots, graphs, or statistical information.

   (Note that Gaussian Naive Bayes can sometimes be effective even if the assumptions are not very reasonable. In order to do correct classification, it is enough to determine the correct MAP class. It is not necessary to actually compute the correct posterior probability $P(C|x)$ for each class.)

What to Hand In

Hand in a zip file which contains all the files needed to run your program, AND the file `proganswers.pdf`. If it is not obvious how to run your program, you should include a README file.

Do not include the `spamtrain.csv, spamtest.csv, and spambase1.txt` in your zip file.