10-315 Intro to Machine Learning HW1

INSTRUCTIONS

- **Due: Monday, September 23 2020 at 11:59 PM EDT.**
- **Format:** Complete this pdf with your work and answers. Whether you edit the latex source, use a pdf annotator, or hand write / scan, make sure that your answers (tex'ed, typed, or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 10-315, click on the appropriate *Written* assignment, and upload your pdf containing your answers. Don't forget to submit the associated *Programming* component on Gradescope if there is any programming required.
- **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 the course site for more information.

Name	
Andrew ID	
Hours to complete (both written and programming)?	

For staff use only

Q1	Q2	Q3	Q4	Total
/14	/20	/ 26	/ 40	/ 100

Q1. [14 pts] MLE

Consider the following distribution with parameters k and α:

$$p(x \mid k, \alpha) = \begin{cases} \frac{\alpha k^{\alpha}}{x^{\alpha+1}} & x \in [k, \infty) \\ 0 & \text{otherwise} \end{cases}$$

We also have that $k \in (0, \infty)$ and $\alpha \in (0, \infty)$.

This distribution is often used for modeling the distribution of wealth in society, fitting the trend that a large portion of wealth is held by a small fraction of the population. This is due to its nature as a skewed, heavy-tailed distribution (notice that the probability decays polynomially in value of the random variable x, unlike Gaussian, exponential, Laplace or Poisson distributions where the probability decays exponentially in the value of the random variable).

Suppose you have a dataset $\mathcal{D}$ which contains N i.i.d samples x_1,x_2,...,x_N drawn from the above distribution.

(a) [3 pts] Derive the log-likelihood $\ell(k, \alpha; \mathcal{D})$.

$\ell(k, \alpha; \mathcal{D})$**:**

(b) [8 pts] Give the MLE for the parameter α, assuming parameter k is fixed.

$\hat{\alpha}_{MLE}$**:**

(c) [3 pts] Next, give the MLE for the parameter k.

Hint: You may be tempted to set k to infinity, but when $k = \infty$, what happens to $p(x \mid k, \alpha)$?

$\hat{k}_{MLE}$**:**

Q2. [20 pts] Maximum A Posteriori Estimation

We've seen in lecture how generative classifiers like Naive Bayes use information about the distributions of the variables in the training data to derive an optimal classifier. MLE is commonly used for estimating the parameters of these distributions since it is generally easy to compute and behaves well asymptotically (i.e. with lots of data). However, using MLE can be problematic when we don't have enough data to get good parameter estimates. When this happens, we need some other method of estimating parameters that doesn't require lots of training data. This is where MAP estimation can be useful since MAP combines a prior assumption about the underlying distribution with the given training data to estimate parameters.

(a) [2 pts] MAP definition.

Let D be some dataset with rows we assume are conditionally independent and are taken from some distribution with parameter θ. By definition, we know that $\theta_{MAP} = \text{argmax}_\theta \, P(\theta|D)$. Use this definition to derive the equivalent definition $\theta_{MAP} = \text{argmax}_\theta \, P(D|\theta)P(\theta)$, justifying each step.

Solution:

(b) [8 pts] MAP Estimation for Bernoulli random variables.

Consider a dataset D composed of the outcomes of independent coin flips using an unbiased coin. Taking each coin flip to be a Ber(θ) random variable, we can use MAP to estimate the coin's bias θ. We'll use as a prior $\theta \sim \beta(x, y)$. Here, the prior represents pseudo-observations; we haven't observed these outcomes, but they reflect our prior beliefs about the bias of the coin.

Our goal is to consider what happens as we vary (1) the size of D and (2) the correctness of our prior. Compute both the MAP and MLE estimates of θ under the following contexts: [You can use the MAP and MLE expressions from lecture slides.]
(a) $D = \{0\,H,\ 2\,T\},\ \theta \sim \beta(4,4)$
(b) $D = \{7\,H,\ 3\,T\},\ \theta \sim \beta(4,4)$
(c) $D = \{15\,H,\ 11\,T\},\ \theta \sim \beta(4,4)$
(d) $D = \{18\,H,\ 16\,T\},\ \theta \sim \beta(5,3)$
(e) $D = \{40\,H,\ 40\,T\},\ \theta \sim \beta(5,3)$
(f) Give a short (1 - 2 sentence) summary of your findings regarding how our choice of prior affects the estimate for θ, given that our underlying belief is that the coin is fair (for (d) and (e) consider the effect an incorrect prior has on our estimate for θ).

Solution:

(c) [10 pts] MAP Estimation for Gaussian means.
Suppose we have a random variable $X \sim \mathcal{N}(\theta, \sigma^2)$ with i.i.d. observations $x_1, ..., x_n$. We want to estimate θ but are not confident we have enough data to get a good estimate using MLE. So, we'll use MAP. Assume a prior distribution for θ that $\theta \sim \text{Exp}(\mu)$ (recall the density function for an exponential random variable with parameter λ is given by $f_X(x) = \lambda e^{-\lambda x}$). Use as a dataset $D = (x_1, ..., x_n)$.
(a) Use the given prior to compute the MAP estimate for θ. Justify your steps.
(b) Show why this estimate maximizes the a posteriori probability instead of minimizing it.
(c) Use limits to show what happens to the influence of the prior on our estimate of θ as the number of observations goes to infinity.

Solution:

Q3. [26 pts] Naive Bayes

Consider a simple learning problem of determining whether Alice and Bob will go to hiking, where $\mathbf{Y} : Hike \in \{T, F\}$ given the weather conditions $\mathbf{X}_1 : Sunny \in \{T, F\}$, and $\mathbf{X}_2 : Windy \in \{T, F\}$ by a Naive Bayes classifier. Using training data, we estimated the parameters $P(Hike = T) = 0.5$, $P(Sunny = T|Hike = T) = 0.8$, $P(Sunny = T|Hike = F) = 0.7$, $P(Windy = T|Hike = T) = 0.4$ and $P(Windy = T|Hike = F) = 0.5$. Assume that the *true* distribution of $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{Y}$ satisfies the Naive Bayes assumption of conditional independence with the above parameters.

(a) [2 pts] Assume $\mathbf{X}_1 : Sunny$ and $\mathbf{X}_2 : Windy$ are truly independent given $Hike$. Write down the Naive Bayes decision rule for this problem using *both* $\mathbf{X}_1$ and $\mathbf{X}_2$ as features.

Solution:

(b) [8 pts] Given the decision rule above, write down $\mathbf{P}(\mathbf{X_1}, \mathbf{X_2}|\mathbf{Y})$ and the Naive Bayes decision for each setting of the weather conditions in the table below:

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{Y}$	$\mathbf{P}(\mathbf{X_1}, \mathbf{X_2}\|\mathbf{Y})$	$\mathbf{f}(\mathbf{X_1}, \mathbf{X_2})$
F	F	F		
F	F	T		
F	T	F		
F	T	T		
T	F	F		
T	F	T		
T	T	F		
T	T	T		

Table 1: $\mathbf{Y}$ is the true decision, while $f(X_1, X_2)$ is the decision made by Naive Bayes classifier.

Solution:

(c) [2 pts] What is the estimated error rate i.e. $\mathbf{P}(\mathbf{f}(\mathbf{X_1}, \mathbf{X_2}) \neq \mathbf{Y})$ for the Naive Bayes classifier using these two features?

Solution:

Next, suppose we gather more information about weather conditions and introduce a new feature denoting $\mathbf{X}_3$: *Rainy* $\in \{T, F\}$. Assume that each day the weather can be **either** *Rainy* **or** *Sunny*. That is, it can not be both *Sunny* **and** *Rainy* (similarly, it can not be not *Sunny* **and** not *Rainy*).

(d) [2 pts] In the above new case, are any of the Naive Bayes assumptions violated? Why or why not?

Solution:

(e) [8 pts] Given the decision rule above, write down $\mathbf{P(X_1, X_2, X_3|Y)}, \mathbf{P(X_1|Y)P(X_2|Y)P(X_3|Y)}$ and the Naive Bayes decision for each setting of the weather conditions in the table below. Notice that when calculating the Naive Bayes prediction $f(X_1, X_2, X_3)$, any violations of the Naive Bayes assumption are ignored.

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{Y}$	$\mathbf{P(X_1, X_2, X_3|Y)}$	$\mathbf{P(X_1|Y)P(X_2|Y)P(X_3|Y)}$	$\mathbf{f(X_1, X_2, X_3)}$
F	F	F	F			
F	F	F	T			
F	F	T	F			
F	F	T	T			
F	T	F	F			
F	T	F	T			
F	T	T	F			
F	T	T	T			
T	F	F	F			
T	F	F	T			
T	F	T	F			
T	F	T	T			
T	T	F	F			
T	T	F	T			
T	T	T	F			
T	T	T	T			

Table 2: $\mathbf{Y}$ is the true decision, while $f(X_1, X_2, X_3)$ is the decision made by Naive Bayes classifier.

Solution:

(f) [4 pts] What is the estimated error rate when the Naive Bayes classifier uses all *three* features? Does the performance of Naive Bayes improve by observing the new feature *Rainy*? Explain why or why not.

Solution:

Q4. [40 pts] Programming

Welcome to the programming component of this assignment!

This assignment includes an autograder for you to grade your answers on your machine. This can be run with the command:

```
python3 autograder.py
```

The code for this assignment consists of several Python files, some of which you will need to read and understand in order to complete the assignment, and some of which you can ignore. You can download and unzip all the code, data, and supporting files from `hw1_programming.zip`.

Files to Edit: `lgr.py, parser.py, sk_lgr.py, sk_nb.py`

You should submit these files containing your code and comments to the Programming component on Gradescope. Please do not change the other files in this distribution or submit any of our original files other than these files. Please do not change the names of any provided functions or classes within the code, or you will wreak havoc on the autograder.

Report: Many of the sections in this programming assignment will contain questions that are not autograded. You will place the requested results in the appropriate locations within the PDF of the Written component of this assignment.

Evaluation: Your assignment will be assessed based on your code, the output of the autograder, and the required contents in the Written component.

Academic Dishonesty: We will be checking your code against other submissions in the class for logical redundancy. If you copy someone else's code and submit it with minor changes, we will know. These cheat detectors are quite hard to fool, so please don't try. We trust you all to submit your own work only; please don't let us down. If you do, we will pursue the strongest consequences available to us.

Getting Help: You are not alone! If you find yourself stuck on something, contact the course staff for help. Office hours, recitation, and Piazza are there for your support; please use them. If you can't make our office hours, let us know and we will schedule more. We want these assignments to be rewarding and instructional, not frustrating and demoralizing. But, we don't know when or how to help unless you ask.

Implementing Logistic Regression

Objective Function

(a) [4 pts] **Q1 Implementation:** In `lgr.py` implement objective to compute the value of the objective for ℓ_2-regularized logistic regression.

This question will be **autograded**. You may run the following command to run a quick unit test on your Q1 implementation:

```
python3 autograder.py -q Q1
```

Note the autograder generates plots of the objective for different values of the regularization term λ.

(b) [4 pts] Describe the effect of the regularization parameter λ on the objective. Attach two plots generated by the autograder for $Q1$ to support your claim and label the axes.

Solution:

Gradient Descent

(c) [6 pts] **Q2 Implementation:** In the gradient_descent function in lgr.py, implement gradient descent for ℓ_2-regularized logistic regression. This question will be **autograded**. You may run the following command to run a quick unit test on our Q2 implementation:

```
python3 autograder.py -q Q2
```

(d) [4 pts] Describe the effect of the learning rate α on convergence. Attach the plot generated by the autograder for $Q2$.

Solution:

Comparing Logistic Regression and Naive Bayes

To compare convergence rates and accuracies of naive bayes and logistic regression you will be using the data in training_set.txt and testing_set.txt. In the provided input files you will find a processed collection of old applications for credit cards along with a corresponding label stating whether they were accepted or rejected by the bank. The goal in this problem is to learn the underlying function of these decisions and be able to classify (label) any new application as *accepted* or *rejected*.

In the data files, each line (credit card application data) consists of 16 values separated by a comma ",". The first 15 values represent attribute values and the last point is the corresponding label for that instance data point. Table 3 gives more information about each attribute in terms of the type and values it can take. In each of the data files, all attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. The features are a mix of multi-nominal and continuous attributes.

i	n_i	Possible Values
00	2	b, a
01	-	continuous
02	-	continuous
03	4	u, y, l, t
04	3	g, p, gg
05	14	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
06	9	v, h, bb, j, n, z, dd, ff, o
07	-	continuous
08	2	t, f
09	2	t, f
10	-	continuous
11	2	t, f
12	3	g, p, s
13	-	continuous
14	-	continuous
label	2	"+"/"-"

Table 3: Description of attribute information in provided Dataset 2, i denotes the attribute and n_i the number of values it takes.

Note the files are comma separated and the last character of each line is a binary class value(0 or 1).

Data Parsing

Some of the data is categorical. However for logistic regression in particular we seek to encode it numerically. To do this we have two immediate options: encoding as an integer or as a *one-hot vector*. Suppose we have a categorical variable distributed over n discrete values $\{m_1, ..., m_n\}$. Then the one-hot vector encoding of m_i is the ith standard basis vector $e_i \in \mathbb{R}^n$.

(e) [2 pts] How should we encode a categorical variable

- Using a one-hot encoding n-vector for n possible categorical values
- Using a discrete scalar variable ranging from 1 to n for n possible categorical values

Why?

Solution:

(f) [4 pts] **Q3 Implementation:** Now implement file_reader in parser.py. Note: this will be somewhat tedious, yet much of data science is working with/manipulating data sets. This question will be **autograded**. To run the autograder on this question we have the command:

```
python3 autograder.py -q Q3
```

Comparison

Now we seek to to compare the accuracies of gaussian naive bayes and logistic regression on varying sizes of data. To do so we will be using the widely used implementations of both from the sk_learn library, to have a common platform.

Implementation: Implement lgr in sk_lgr.py, gnb in sk_gnb.py, using the sklearn library. Put the code plotting in plotting.py. We encourage you to employ good practice and familiarize yourself with the relevant parts of the library(involving logistic regression and naive bayes). The framework provided allows us to incorporate priors and regularizes, but we will simply be running code with the default parameters. Note we ask you to train/run a model 10 times for a specified training set size.

Note this will **not be** autograded but instead graded solely on figures and analysis.

(g) [4 pts] Plot together testing accuracies for gnb and lgr over subsets of the dataset of size ranging over $[.1, .2, .3, .4, .5, .6, .7, .8, .9, 1.0]$. Attach pictures of the testing accuracy plot and training accuracy plot and label them.

Plot:

(h) [4 pts] Which model seems to outperform the other? Is this what we would expect on small amounts of data? Why/why not?

Plot:

Implementation: Now implement lgr_repeated and gnb_repeated in sk_lgr.py, sk_gnb.py. This should be nearly exactly the same as previously except all the features in each feature vector should be duplicated(so 94 features instead of 47).

(i) [4 pts] Plot together testing accuracies for gnb_repeated and lgr_repeated over subsets of the dataset of size ranging over $[.1, .2, .3, .4, .5, .6, .7, .8, .9, 1.0]$. Attach pictures of the testing accuracy plot and training accuracy plot, and label them.

Plot:

(j) [4 pts] How does training/testing compare to the non duplicated case? Which case would we prefer computationally and why? Does there seem to be any advantage to keeping highly correlated features in training?

Plot:

Submission

Complete all questions as specified in the above instructions. Then upload lgr.py, parser.py, sk_gnb.py, sk_lgr.py, and plotting.py to Gradescope(do not zip). Your submission should finish running within 10 minutes, after which it will time out on Gradescope.

Don't forget to include any request results in the PDF of the Written component, which is to be submitted on Gradescope as well.

You may submit to Gradescope as many times as you like. You may also run the autograder on your own machine to speed up the development process. Just note that the autograder on Gradescope will be slightly different than the local autograder. The autograder can be invoked on your own machine using the command:

```
python3 autograder.py
```

Note that running the autograder locally will not register your grades with us. Remember to submit your code when you want to register your grades for this assignment.

The autograder on Gradescope might take a while but don't worry: so long as you submit before the deadline, it's not late.

Collaboration Questions
After you have completed all other components of this assignment, report your answers to the collaboration policy questions detailed in the Academic Integrity Policies found on the course site.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.

2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details?

3. Did you find or come across code that implements any part of this assignment ? If so, include full details.