CSCI4360/6360 - HW2

Instructor: Dr. Ninghao Liu (ninghao.liu@uga.edu)

February 1, 2024

– Upload two files to eLC (failing to do so will receive 20% penalty):
1) a scanned handwritten solution or typed pdf file named "*MyID_HW2.pdf*" containing your answer to each question and code running results;
2) a zip file named "*MyID_HW2PQ.zip*" containing your programs for Question 1.

– Due Date: Feb 21, 2024

1 Linear Model (40 pts)

1) (10 pts) Suppose we are working on a regression problem, and we want to use a linear model to solve the problem. The training data is denoted as $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, where $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, $\boldsymbol{y} \in \mathbb{R}^{N}$, N is the number of samples, and D is the feature dimension. We define the linear model as $f(\boldsymbol{x}) = \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}$, where $\boldsymbol{w} \in \mathbb{R}^{D+1}$ and $\boldsymbol{x} \in \mathbb{R}^{D+1}$. We denote the training loss as

$$L(\mathcal{D}, \boldsymbol{w}) = \frac{1}{2} \sum_{(\boldsymbol{x},y) \in \mathcal{D}} (y - f(\boldsymbol{x}))^2 = \frac{1}{2} \sum_{(\boldsymbol{x},y) \in \mathcal{D}} (y - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x})^2. \tag{1}$$

Please write the pseudo code for stochastic gradient decent for training the linear model. Please provide the detailed math formula of gradient computation, as well as how to update $\boldsymbol{w}$.
Note: The dimension of input $\boldsymbol{x}$ is $D+1$, instead of D, meaning that we have done the data augmentation by appending an additional "1" to all the samples. This is also why there is no bias term in our linear model f.
2) (15 pts) Finish the program in "LinearRegression.py". Take a screenshot or write down the lines of code you write. Report the final linear model after training.
3) (15 pts) Finish the program in "LogisticRegression.py". Take a screenshot or write down the lines of code you write. Report the program output.

2 Naive Bayes Classifiers (35 pts)

Given the following dataset (each instance has three features):

No.	Outlook	Temperature	Humidity	Play Golf
1	sunny	hot	high	N
2	sunny	hot	high	N
3	overcast	hot	high	Y
4	rain	mild	high	Y
5	rain	cool	normal	N
6	rain	cool	normal	N
7	overcast	cool	normal	Y
8	sunny	cool	normal	Y
9	sunny	mild	normal	Y
10	sunny	mild	high	?

1) (20 pts) Classify instance No. 10 using Naive Bayes Classifier (NBC). Include the details of your NBC, probability calculations, and how the final classification is determined.

2) (5 pts) What is the time complexity for training and testing Naive Bayes classifier, respectively?

3) (10 pts) After a yearly checkup for a software developer, there are both bad news and good news from the doctor. The bad news is that the developer has a test result positive for a serious disease, and the test is 98% accurate (i.e., if you have the disease, then the probability of testing positive is 0.98; if you do not have the disease, the probability of testing negative is also 0.98). The good news is that this is a rare disease, because only 1 in 20,000 people will have it. What are the chances that the developer actually has the disease?

3 Decision Trees (25pts)

Given the dataset below, where we want to classify whether an social networks account is real or not. Here "Posts", "Friends", "Photo" are the features/attributes, meaning the frequency of writing posts, number of friends, and whether use the real photo, respectively. "Real Account" is the label. Attribute values "S", "M", "L" means a "small", "medium" and "large" number, respectively. Suppose we want to use Information Gain to build a decision tree model (ID3).

No.	Posts	Friends	Photo	*Real Account*
1	S	S	NO	NO
2	S	L	YES	YES
3	L	M	NO	YES
4	M	M	YES	YES
5	L	M	YES	YES
6	M	L	NO	YES
7	M	S	NO	NO
8	L	M	NO	YES
9	M	S	NO	NO
10	S	S	YES	YES

1) (5 pts) Compute the Information Gain if we first choose "Friends" as the attribute to split data.

2) (15 pts) Construct a decision tree from the given data. Show the computation steps.

3) (5 pts) Explain the limitation of using Information Gain as the attribute splitting measure.