ISYE-6740

Homework 3

1. Basic optimization. (30 points.) Consider a simplified logistic regression problem. Given m training samples (xi,yi), i = 1,...,m. The data xi ∈ R (note that we only have one feature for each sample), and yi ∈ {0,1}. To fit a logistic regression model for classification, we solve the following optimization problem, where ϑ ∈ R is a parameter we aim to find:

$$\max \mathrm{l}(\Theta)$$

where the log-likelihood function:

$$\mathrm{l}(\Theta) = \sum_{i=1}^{m} \{ -\log(1 + \exp\{\Theta x_i\}) + (y_i - 1)\Theta x_i \}$$

(a) (10 points) Show step-by-step mathematical derivation for the gradient of the cost function `(ϑ) in (1) and write a pseudo-code for performing gradient descent to find the optimizer ϑ∗. This is essentially what the training procedure does. (pseudo-code means you will write down the steps of the algorithm, not necessarily any specific programming language.)

Step 1) First part in finding the gradient for the cost function is to find the partial derivative for sum of the terms in this equation:

$$\frac{\partial \mathrm{l}(\Theta)}{\partial_j} = \frac{\partial \mathrm{l}(\Theta)}{\partial_j} - \log(1 + \exp\{\Theta x_i\}) + \frac{\partial \mathrm{l}(\Theta)}{\partial_j}(y_i - 1)\Theta x_i\} \tag{1}$$

Step 2) Taking the derivative of log(x) is equal to 1/x then apply the chain rule we get:

$$\mathrm{l}' = \left\{ \frac{x^i e^{-\Theta x_i}}{\left(1 + e^{-\Theta x_i}\right)} + (y_i - 1)x_i \right\} \tag{2}$$

Pseudo-Code

1. Choose step-size value n
2. Initialize initial parameter for coefficients θ_0
3. while not converge and not reach the max iters:

    Multiply step-size n by the gradient ll:

    $\theta = \theta^{t-1} + \eta \cdot \mathrm{l}'\left(\theta^{t-1}\right)$
4. End

b) Stochastic Gradient Descent Pseudo-code:

1. Choose step-size value n
2. Initialize initial parameter for coefficients θ_0, t $= 0$
3. while not converge and t $\leq$ max iters:

   Divide dataset into K subsets, X_k

   for each X_k find θ choose k :

$$l' = \sum i \in X_k \left\{ \frac{x^i e^{-\Theta x_i}}{\left(1 + e^{-\Theta x_i}\right)} + (y_i - 1)x_i \right\}$$

$$\theta^{t}_{k} = \theta^{t-1}_{k-1} + \eta \cdot l' \left(\theta^{t-1}_{k-1}\right)$$

   End for loop

4. End

c) Hessian will become scalar since we're only dealing with one parameter, in our case to determine is our function is concave we will find the second order derivative to determine a global optimum:

$$l' = \left\{ \frac{x^i e^{-\Theta x_i}}{\left(1 + e^{-\Theta x_i}\right)} + (y_i - 1)x_i \right\} \tag{3}$$

$$l'' = \left\{ \frac{x^i e^{-\Theta x_i}\left(1 + e^{-\Theta x_i}\right)}{\left(1 + e^{-\Theta x_i}\right)} + \frac{e^{-\Theta x_i} x^i e^{-\Theta x_i}}{\left(1 + e^{-\Theta x_i}\right)^2} \right\} \tag{4}$$

$$l'' = \left\{ \frac{-x_i^2 e^{-\Theta x_i}}{\left(1 + e^{-\Theta x_i}\right)^2} \right\} < 0 \tag{5}$$

2. Comparing Bayes, Logistic, and KNN Classifiers

Part 1) Divorce classification/prediction

a-c)

Provided is the performance / model accuracy for each of the 3 classifiers. This chart shows the iterations with the number of features used and the dataset used, in our case the divorce and digits dataset.

Model_Name	Model_Accuracy	Num_Features	Dataset
GNB (Gaussian Naive Bayes)	0.97	All	divorce
KNN	0.97	All	divorce
Logistic	0.97	All	divorce

GNB (Gaussian Naive Bayes)	0.94	2	divorce
KNN	0.91	2	divorce
Logistic	0.93	2	divorce
GNB (Gaussian Naive Bayes)	0.56	All	digits
KNN	0.52	All	digits
Logistic	0.52	All	digits

Based on our results we can determine that the training scores after 1 iteration is higher then the scores used for 2 features. However, this is clearly due to the model having only chosen the first 2 features in the dataset, which means we're excluding other features that have high predictive power and account for significant variance in the target variable.

b)

Decision boundary plots for 2-features used in the divorce dataset:

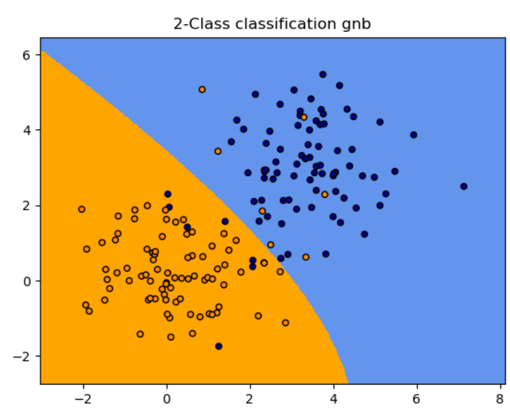

Figure 1: Decision Boundary Plot GNB

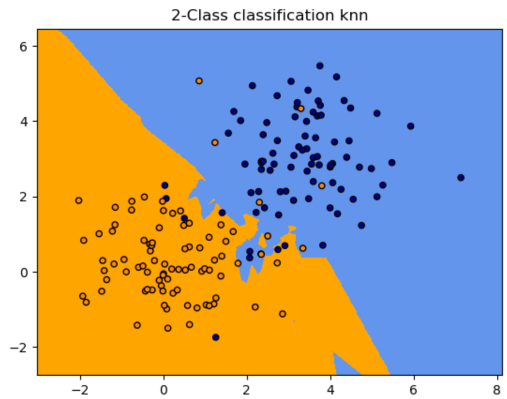

Figure 2: Decision Boundary Plot KNN

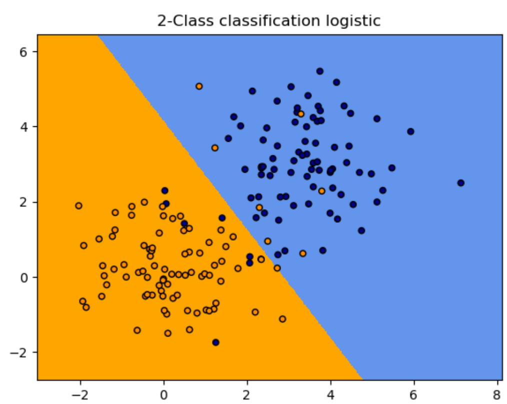

Figure 3: Decision Boundary Plot Logistic

We can see with the decision boundaries that the results are linearly separable, which means our classifiers will be able to equally account for a majority of the variance and have strong accuracy when predicting against our target variable.

3.

a)

We have 3 messages that are spam and 4 non-spam, therefore:

$\mathrm{P}(\mathrm{y} = 0) \;=\; \frac{3}{7} \quad \mathrm{P}(\mathrm{y} = 1) \;=\; \frac{4}{7}$

b)

Spam Message Feature Vectors:

million dollar offer : $[0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$

secret is secret : $[2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]$

secret offer today : $[1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$

Non-Spam Message Feature Vectors:

low price for valued customer : $[0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0]$
play secret sports today : $[1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0]$
sports is healthy : $[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0]$
low price pizza : $[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$

c) Not-complete
d) To calculate the posterior probabilities, we're first going to need to calculate our conditional probabilities based on the probability of each word of the message appearing in a spam message vs a non-spam message.

Prior Probability (y=0) $= \frac{3}{7}$	Prior Probability (y=1) $= \frac{4}{7}$
Conditional Probability $P(\text{"today"}\|y=0) = \frac{1}{9}$ $P(\text{"is"}\|y=0) = \frac{1}{9}$ $P(\text{"secret"}\|y=0) = \frac{3}{9}$ $P(\text{"today is secret"}\|y=0) = \frac{1}{243}$	Conditional Probability $P(\text{"today"}\|y=1) = \frac{1}{15}$ $P(\text{"is"}\|y=1) = \frac{1}{15}$ $P(\text{"secret"}\|y=1) = \frac{1}{15}$ $P(\text{"today is secret"}\|y=1) = \frac{1}{3375}$

$$P(y=0|\text{"today is secret"}) = \frac{P(\text{"today is secret"}|y=0) * P\ (y=0)}{P(\text{"today is secret"}|y=1) * P\ (y=1) + P(\text{"today is secret"}|y=0) * P\ (y=0)}$$

$$P(y=0|\text{"today is secret"}) = \frac{\frac{1}{243} * \frac{3}{7}}{\frac{1}{3375} * \frac{4}{7} + \frac{1}{243} * \frac{3}{7}}$$

$$P(y=0|\text{"today is secret"}) = \frac{\frac{1}{567}}{\frac{137}{70875}}$$

$$P(y=0|\text{"today is secret"}) = \frac{1}{567} * \frac{70875}{137}$$

$$P(y=0|\text{"today is secret"}) = \frac{70875}{77679}$$

$$P(y=0|\text{"today is secret"}) = \frac{125}{137}$$

Therefore, with a probability of 91.24% we can classify this message as spam as it's well above the threshold.