Stony Brook University
CSE512 – Machine Learning – Fall 19
Homework 3, Version 4, updated 3 Oct 2019
Due: 6 Oct 2019 at midnight 23:59

This homework contains 2 questions. The last question requires programming. The maximum number of points is 100 plus 10 bonus points.

1 Question 1 – Nearest Neighbor Classifiers (40 points)

1.1 1-NN with asymmetric loss (20 points)

Suppose we want to build a binary classifier, but the cost of false positive (predicting positive for a negative case) is much higher than the cost of false negative (predicting negative for a positive case). One can consider an asymmetric loss function, where the cost of false negative is 1, while the cost of false positive is $\alpha > 1$. For a point x, let $\eta(x)$ be the probability that x is positive.

1.1.1 (3 points)

Show that the optimal Bayes risk for data point x is $r^*(x) = \min\{\eta(x), \alpha(1-\eta(x))\}$.

1.1.2 (4 points)

Let $r(x)$ be the asymptotic risk for data point x, express $r(x)$ in terms of α and $\eta(x)$.

1.1.3 (10points)

Prove that $r(x) \leq (1+\alpha)r^*(x)(1-r^*(x))$. Hint: use $\alpha > 1$

1.1.4 (3 points)

Let R be the asymptotic risk of the 1-NN classifier and R^* be Bayes risk. Prove that: $R \leq (1+\alpha)R^*(1-R^*)$

1.2 k-NN classifier (20 points)

Consider a k-NN classifier: classify a point as positive if at least (k+1)/2 nearest neighbors are positive.

1.2.1 (3 points)

Consider drawing k points randomly from a Bernoulli distribution with two outcomes: positive or negative, and the probability of the point being positive is η. Let $g(\eta, k)$ be the probability that at least $(k+1)/2$ out of k points are positive. Express the asymptotic risk $r(x)$ for a point x in terms of $\eta(x)$ and the function $g(\cdot, \cdot)$.

1.2.2 (10 points)

Prove that $r(x) = r^*(x) + (1 - 2r^*(x))g(r^*(x), k)$

1.2.3 (3 points)

Using Hoeffding's Inequality (`https://en.wikipedia.org/wiki/Hoeffding_inequality`), prove that:

$$g(r^*(x), k) \leq \exp(-2(0.5 - r^*(x))^2 k) \tag{1}$$

1.2.4 (4 points)

Prove that: $r(x) \leq r^*(x) + \frac{1}{\sqrt{2k}}$. Hint: you should use the above inequality Eq. (1). Note that: from this result, you can see that the Asymptotic risk of k-NN classifier is the Bayes Risk if k goes to infinity.

2 Question 2 – Implementation of Logistic Regression Classifier for k classes (60 points + 10 bonus)

In this Question, you will implement Logistic Regression using Stochastic Gradient Descent (SGD) optimization. Suppose the training data is $\{(X^1, Y^1), \cdots, (X^n, Y^n)\}$, where X^i is a column vector of d dimensions and Y^i is the target label. For a column vector X, let $\bar{X}$ denotes $[X; 1]$, the vector obtained by appending 1 to the end of X. $\boldsymbol{\theta}$ is the set of parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{k-1}$. Logistic regression for k classes assumes the following probability function:

$$P(Y = i|X; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}_i^T \bar{X})}{1 + \sum_{j=1}^{k-1} \exp(\boldsymbol{\theta}_j^T \bar{X})} \text{ for } j = 1, \cdots, k-1, \tag{2}$$

$$P(Y = k|X; \boldsymbol{\theta}) = \frac{1}{1 + \sum_{i=1}^{k-1} \exp(\boldsymbol{\theta}_j^T \bar{X})}. \tag{3}$$

Logistic regression minimizes the average conditional log likelihood:

$$L(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \log(P(Y^i|\bar{X}^i; \boldsymbol{\theta})). \tag{4}$$

To minimize this loss function, we can use gradient descent:

$$\boldsymbol{\theta}_c = \boldsymbol{\theta}_c - \eta \frac{\partial L}{\partial \boldsymbol{\theta}_c} \tag{5}$$

$$\text{where } \frac{\partial L}{\partial \boldsymbol{\theta}_c} = -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial \log(P(Y^i|\bar{X}^i; \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_c} \tag{6}$$

This gradient is computed by enumerating over all training data. It turns out that this gradient can be approximated using a batch of training data. Suppose $\mathcal{B}$ is a subset of $\{1, 2, \cdots, n\}$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_c} \approx -\frac{1}{card(\mathcal{B})} \sum_{i \in \mathcal{B}} \frac{\partial \log(P(Y^i|\bar{X}^i; \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_c} \tag{7}$$

This leads to the following stochastic gradient descent algorithm:

Algorithm 1 Stochastic gradient descent for Logistic Regression

1: Inputs: $\{(X_i, Y_i)\}_{i=1}^n$ (for data), m (for batch size), η_0, η_1 (for step size), max_epoch, δ (stopping criteria)
2: **for** $epoch = 1, 2, ..., max\_epoch$ **do**
3: $\quad \eta \leftarrow \eta_0/(\eta_1 + epoch)$
4: $\quad (i_1, ..., i_n) = permute(1, ..., n)$
5: $\quad$ Divide $(i_1, ..., i_n)$ into batches of size m or $m + 1$
6: $\quad$ **for** each batch $\mathcal{B}$ **do**
7: $\quad\quad$ `Update` $\boldsymbol{\theta}$ `using Eqs. (5) and (7)`
8: $\quad$ Break if $L(\boldsymbol{\theta}^{new}) > (1 - \delta)L(\boldsymbol{\theta}^{old})$ // not much progress, terminate
9: Outputs: $\boldsymbol{\theta}$.

2.1 Derivation (10 points)

Prove that:

$$\frac{\partial \log(P(Y^i|\bar{X}^i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_c} = (\delta(c = Y^i) - P(c|\bar{X}^i;\boldsymbol{\theta}))\bar{X}^i. \tag{8}$$

where $\delta(c = Y^i)$ is the indicator function which takes a value of 1 if the class c equals the ground truth label Y^i, and 0 otherwise. Use Equation (8) to derive the gradient of the loss with respect to the parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{k-1}$.

2.2 Crowd Image Classification

In this question of the homework, you will work with image data. We are providing you with the features extracted from the crowd images, so you do not need to extract features from the raw images. We are also providing you with the raw images for error analysis and visualization purposes. Your task is to classify an image into 4 different categories. The data has already been split into train, validation, and test sets.

Dataset details (obtain data from Kaggle competition page)
—- train set (4000 images)
—- val set (2000 images)
—- test set (2000 images)

2.3 Implement Logistic Regression with SGD (50 points + 10 bonus)

Your task is to implement Logistic Regression with $k = 4$ classes using SGD. You should use Python or Matlab for your implementation.

1. (15 points) Run your implementation on the provided training data with $max\_epoch = 1000, m = 16, \eta_0 = 0.1, \eta_1 = 1, \delta = 0.00001$.
   (a) Report the number of epochs that your algorithm takes before exiting.
   (b) Plot the curve showing $L(\boldsymbol{\theta})$ as a function of $epoch$.
   (c) What is the final value of $L(\boldsymbol{\theta})$ after the optimization?
2. (10 points) Keep $m = 16, \delta = 0.00001$, experiment with different values of η_0 and η_1. Can you find a pair of parameters (η_0, η_1) that leads to faster convergence?
   (a) Report the values of (η_0, η_1). How many epochs does it take? What is the final value of $L(\boldsymbol{\theta})$?
   (b) Plot the curve showing $L(\boldsymbol{\theta})$ as a function of $epoch$.
3. (10 points) Evaluate the performance on validation data
   (a) Plot $L(\boldsymbol{\theta})$ as a function of $epoch$. On the same plot, show two curves, one for training and one for validation data.
   (b) Plot the accuracy as a function of $epoch$. On the same plot, show two curves, one for training and one for validation data.
4. (5 points) With the learned classifier:
   (a) Report the confusion matrices on the validation and the training data.

2.4 Submit the result in Kaggle (10 points + 10 bonus)

1. (10 points) Run your classifier on the test data and submit the result file on Kaggle (`https://www.kaggle.com/c/cse512hw3`). Report the best accuracy you obtain on the test data, as returned by the Kaggle website.

2. (10 bonus points) Optimize your classifier and try feature engineering to compete for the leading positions as the top 3 positions will receive the bonus points! You can make use of concepts taught in class such as regularization, feature normalization to achieve better performance. You can use validation data to tune your classifier. You can use validation data together with the training data to train your classifier, once you've identified the best combination of hyper-parameters, to generate test predictions for Kaggle submission. You can be creative with the construction of feature vectors. But you cannot use deep learning CNN features of the images.

NOTE: Make sure the names displayed on the leader-board (your Kaggle user-names) can be used by the graders to uniquely identify you (SBU ID is recommended).

3 What to submit?

3.1 Blackboard submission

You will need to submit both your code and your answers to questions on Blackboard. Put the answer file and your code in a folder named: SUBID_FirstName_LastName (e.g., *10947XXXX_lionel_messi*). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file, i.e, SUBID_FirstName_LastName.zip. The answer file should be named: answers.pdf. The first page of the answers.pdf should be the filled cover page at the end of this homework. The remaining of the answer file should contain:

1. Answers to Question 1.1 and 1.2
2. Answers to Question 2.1 and the requested plots, numbers and confusion matrices for 2.3 and 2.4.
3. For 2.3, also submit separate Python/Matlab code for each of the 4 sub-parts.
4. For 2.4, write the accuracy from kaggle website.

3.2 Prediction submission

For Question 2.4, you must submit a `.csv` file to get the accuracy through Kaggle. A submission file should contain two columns: Id and Class. The file should contain a header and have the following format:

Id,	$Class$
1,	1
2,	10
...	...

You can only make **3 submissions per day**. In the end, your top-2 entries (you can manually choose them) will be evaluated on a held-out division of the test set. The final rankings will be the ones as obtained through this private leader-board.

4 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You must use your SBU ID as your file name for the competition. Do not fake your Stony Brook ID to bypass the submission limitation per 24 hours. Doing so will be considered cheating.

Cover page for answers.pdf
CSE512 Fall 2019 - Machine Learning - Homework 3

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: