Stony Brook University
CSE512 – Machine Learning – Fall 18
Homework 3, Version 1
Due: 28 Sep 2018 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 – Naive Bayes and Logistic Regression (40 points)

1.1 Naive Bayes with both continuous and boolean variables (20 points)

Consider learning a function $\mathbf{X} \rightarrow Y$ where Y is boolean, where $\mathbf{X} = (X_1, X_2)$, and where X_1 is a boolean variable and X_2 a continuous variable. State the parameters that must be estimated to define a Naive Bayes classifier in this case. Give the formula for computing $P(Y|\mathbf{X})$, in terms of these parameters and the feature values X_1 and X_2.

1.2 Naive Bayes and Logistic Regression with Boolean variables (20 points)

In class, we showed that when Y is Boolean and $\mathbf{X} = (X_1, \cdots, X_d)$ is a vector of continuous variables, the the assumptions of the Gaussian Naive Bayes classifier imply that $P(Y|\mathbf{X})$ is given by the logistic function with appropriate parameters $\boldsymbol{\theta}$. In particular:

$$P(Y = 1|\mathbf{X}) = \frac{1}{1 + \exp(-(\sum_{i=1}^{d} \theta_i X_i + \theta_{d+1}))} \tag{1}$$

Consider instead the case where Y is Boolean and $\mathbf{X} = (X_1, \cdots, X_d)$ is a vector of *Boolean* variables. Prove for this case also that $P(Y|\mathbf{X})$ follows this same form (and hence that Logistic Regression is also the discriminative counterpart to a Naive Bayes generative classifier over Boolean features).

2 Question 2 – Implementation of Logistic Regression (60 points + 10 bonus)

In this Question, you will implement Logistic Regression using Stochastic gradient descent 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 binary target label. For a column vector X, let $\bar{X}$ denotes $[X; 1]$, the vector obtained by appending 1 to the end of X. Logistic regression assumes the following probability function:

$$P(Y = 1|X; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \bar{X})} \tag{2}$$

Logistic regression minimizes the negative conditional log likelihood:

$$-\sum_{i=1}^{n} \log(P(Y^i|\bar{X}^i; \boldsymbol{\theta})) \tag{3}$$

An equivalent minimization problem is to minimize 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:

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

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

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 $\{1, 2, \cdots, n\}$

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

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:

$$\frac{\partial \log(P(Y^i|\bar{X}^i; \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} = (Y^i - P(Y = 1|\bar{X}^i; \boldsymbol{\theta}))\bar{X}^i \tag{8}$$

2.2 Hand Classification

In this question of the homework, you will work with image data. Your task is to classify between hand/not-hand images. The data has already been split into train, validation, and test sets. For the training and validation sets, the first half of the images are hands and second half are not-hands.

Raw image intensity values are not robust features for classification. In this question, we will use Histogram of Oriented Gradient (HOG) as image features. HOG uses the gradient information instead of intensities, and this is more robust to changes in color and illumination conditions. See [1] for more information about HOG.

We will implement this question in Python. To use HOG, you will need to install skimage. This is an excellent image processing library. In fact, you should not call HoG directly. Use the supplied helper function load_dataset() instead; it will call HoG when loading the dataset.

The provided homework directory has the following structure:
root
– hw-3.pdf
– logistic_regression.ipynb
– requirements.txt
– data (obtain from Kaggle competition page)
—- train (8170 images, 0-4084.png: hand, 4085-8169.png: not-hand)
—- val (2724 images, 0-1361.png: hand, 1362-2723.png: not-hand)
—- test (5542 images)

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

We have provided the skeleton code in the form of a Jupyter Notebook. Your task is to complete the implementations in it.

1. (10 points) Run your implementation on the provided training data with $max\_epoch = 1000, m = 16, \eta_0 = 0.1, \eta_1 = 1, \delta = 0.0001$.
   (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 $m = 16, \delta = 0.0001$, 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. (10 points) With the learned classifier: (You can plot these with sklearn in python, simply install it by: pip install scikit-learn)
   (a) Plot the ROC curve on validation data. Report the area under the curve.
   (b) Plot the Precision-Recall curve on validation data. Report the average precision.

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/t/dbe3655b7c01457d8a92348881887dd1`). Report the best accuracy you obtain on the test data, as returned by the Kaggle website.
2. (10 bonus points) Optimize your classifier to compete for the leading positions as the top 3 positions will receive the bonus points! 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 and numbers for 2.3 and 2.4.
3. For 2.3, the completed .ipynb file with the numbers and plots as asked for in the several sub-questions. If any explanation asked, include that in the notebook in the form of a Markdown cell.
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.

5 Reference

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.

Cover page for answers.pdf
CSE512 Fall 2018 - Machine Learning - Homework 3

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: