Fall 2022 CS165B: Introduction to Machine Learning – Homework 3

Due: **Sunday, Nov 13th, 11:59 pm PST**

Note: Please upload your written part as a PDF, and please upload your code as a separate file in (".py" or ".ipynb") form. Both files should be uploaded to Gauchospace.

1. (25 points) Cross-entropy error measure.

   (a) (12 points) If we are learning from ± 1 data to predict a noisy target $P(y|\mathbf{x})$ with candidate hypothesis h, show that the maximum likelihood method reduces to the task of finding h that minimizes

   $$E_{in}(\mathbf{w}) = \sum_{n=1}^{N} [\![y_n = +1]\!] \ln \frac{1}{h(\mathbf{x}_n)} + [\![y_n = -1]\!] \ln \frac{1}{1 - h(\mathbf{x}_n)}$$

   Hint: Use the likelihood $p(y|x) = \begin{cases} h(x) & \text{for } y = +1 \\ 1 - h(x) & \text{for } y = -1 \end{cases}$ and derive the maximum likelihood formulation.

   (b) (13 points) For the case $h(\mathbf{x}) = \theta\left(\mathbf{w}^T \mathbf{x}\right)$, argue that minimizing the in-sample error in part (a) is equivalent to minimizing the one given below

   $$E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)$$

   Note: For two probability distributions $\{p, 1-p\}$ and $\{q, 1-q\}$ with binary outcomes, the cross-entropy (from information theory) is

   $$p \log \frac{1}{q} + (1-p) \log \frac{1}{1-q}.$$

   The in-sample error in part (a) corresponds to a cross-entropy error measure on the data point $(\mathbf{x}_n, y_n)$, with $p = [\![y_n = +1]\!]$ and $q = h(\mathbf{x}_n)$.

2. (25 points) For logistic regression, show that

   $$\begin{aligned} \nabla E_{in}(\mathbf{w}) &= -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}} \\ &= \frac{1}{N} \sum_{n=1}^{N} -y_n \mathbf{x}_n \theta\left(-y_n \mathbf{w}^T \mathbf{x}_n\right) \end{aligned}$$

   Argue that a "misclassified" example contributes more to the gradient than a correctly classified one.

   Hint: Recall the logistic regression objective function $E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + \exp\left(-y_n \mathbf{w}^T \mathbf{x}_n\right)\right)$ and take it's derivative with respect to $\mathbf{w}$.

3. (20 points) Handwritten Digits Data: You should download the data files with handwritten digits data including only 1 and 5: training data (train_data.npy), training labels (train_labels.npy), test data (test_data.npy), and test labels (test_labels.npy). You can use np.load() to load the npy files. Each row of train_data and test_data represents one data point. train_data should be a 1561×256 matrix and test_data should be a 424×256 matrix. Each data point has 256 gray scale values between -1 and 1. The 256 pixels correspond to a 16×16 image. train_labels and test_labels are 1561 and 424 dimensional arrays respectively, and they have label 1 for digit 1 and label -1 for the digit 5.

   (a) (5 points) Plot two of the digit images, one for digit 1 and one for digit 5.

   (b) (10 points) Extract two features to distinguish 1 and 5. For example, you may use symmetry and average intensity. You can also use other features defined by yourself.

   (c) (5 points) Provide 2-D scatter plots of your features for training and test data (Now your data matrix will be N×2). For each data example, plot the two features with a red $\times$ if it is a 5 and a blue $\circ$ if it is a 1.

4. (30 points) Classifying Handwritten Digits: 1 vs. 5. Implement logistic regression for classification using gradient descent to find the best linear separator you can using the training data only (use your 2 features from the above question and an additional 1 as the inputs). The output is +1 if the example is a 1 and -1 for a 5.

   (a) (6 points) Give separate plots of the training and test data, together with the separators (Similar to what you did in PLA homework. After you learn the model vector w, you can plot a line).

   (b) (6 points) Compute train E_{in} and test E_{test} errors. Use only the training data to compute training error and use only the test data to compute the test error.

   (c) (6 points) Logistic regression can also have regularization: $\min_{\mathbf{w}} E(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$, where $E(\mathbf{w})$ is the logistic loss. Change your gradient descent algorithm accordingly and repeat (b). Report the best λ using cross-validation.

   (d) (6 points) Now repeat (b) using a 3rd order polynomial transform.

   (e) (6 points) As your final deliverable to a customer, would you use the linear model with or without the 3rd order polynomial transform? Explain.