CSE 575: Statistical Machine Learning Assignment #2

Instructor: Prof. Hanghang Tong
Out: Sep. 15th, 2018; Due: Oct. 18th, 2018

Submit electronically, using the submission link on Blackboard for Assignment #2, a file named `yourFirstName-yourLastName.pdf` *containing your solution to this assignment (a .doc or .docx file is also acceptable, but .pdf is preferred).*

1 Logistic Regression [15 points]

Now we consider the following two logistic regression models:

- Model 1: $P(Y = 1 \mid \mathbf{X}, w_1, w_2, w_3) = \frac{1}{1+exp(w_1X_1+w_2X_2+w_3X_3)}$
- Model 2: $P(Y = 1 \mid \mathbf{X}, w_0, w_1, w_2, w_3) = \frac{1}{1+exp(w_0+w_1X_1+w_2X_2+w_3X_3)}$

Suppose we have three training examples:

$$\mathbf{X}^{(1)} = [1,\ 1,\ 1]^T \quad \mathbf{X}^{(2)} = [1,\ 1,\ 0]^T \quad \mathbf{X}^{(3)} = [0,\ 0,\ 0]^T$$
$$Y^{(1)} = -1 \qquad Y^{(2)} = -1 \qquad Y^{(3)} = 1$$

Answer the following questions.

- [2 points] How many independent parameters for Model 1? What are they?
- [2 points] How many independent parameters for Model 2? What are they?
- [4 points] Does it matter how the third example is labeled in Model 1? Does it matter in Model 2? Please justify your answers.
- [7 points] For Model 1, we want to maximize the conditional log-likelihood with an L_2 regularization term (we use λ as the penalized parameter). What would the conditional log-likelihood term behave when λ is very large? Can you derive the MLE of the parameter $\mathbf{w}$ in terms of λ and the training data $\mathbf{X}^{(1)}$, $Y^{(1)}$. Based on this, can you explain how the weights $\mathbf{w}$ will behave as λ increases.

2 The Convexity of Logistic Regression [5 points]

Prove that the log conditional likelihood function $l(w)$ in logistic regression is concave. [**Hints:** To ease the question, you can assume that each data has only one attribute and hence w is 1-dimensional.]

3 1NN-Classifier Decision Boundary [20 points]

Given a negative example at $(0, 1)$ and a positive example at $(1, 0)$. We want to train a 1NN classifier.

- [8 points] Prove that the decision boundary of 1NN classifier with L_1 distance is the one as shown in Figure 1 (i.e., we classify all the examples from the green area as positive; all the examples from the red area as negative; and the brown area is the decision boundary).

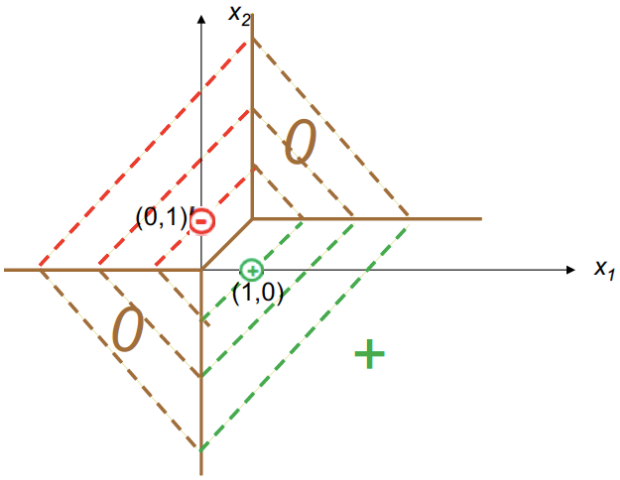

Figure 1: Decision Boundary of 1NN Classifier with L_1 distance metric.

- [4 points] What will be the decision boundary if we use L_2 distance instead?
- [4 points] What will be the decision boundary if we use L_0 distance instead?
- [4 points] What will be the decision boundary if we use L_∞ distance instead?

4 The Margin of SVM [5 points]

In the hard-margin linear SVM formulation, we seek a linear classifier $w' \cdot x + b = 0$, subject to $y_i(w' \cdot x_i + b) \geq 1$ $(i = 1, ..., n)$, where $y_i = \pm 1$ is the class label and x_i is the feature for the i^{th} example, respectively. Prove that the size of the margin of such a classifier $m = \frac{2}{\|w\|}$.

5 Kernelized SVM [15 points]

Given the following four training examples in 2-d space: $X^1 = (-1, -1)$, $y^1 = +1$; $X^2 = (-1, +1)$, $y^2 = -1$; $X^3 = (+1, -1)$, $y^3 = -1$; and $X^4 = (+1, +1)$, $y^4 = +1$. In this exercise, we use the superscript to denote different examples and the subscribe to denote two different features of each example.

- [3 points] For the feature vector of each example, we use the following function to map it to a 6-d feature: $\phi(X^i) = [1, (X_1^i)^2, \sqrt{2}X_1^i X_2^i, (X_2^i)^2, \sqrt{2}X_1^i, \sqrt{2}X_2^i]'$ $(i = 1, 2, 3, 4)$. For this feature map, what is the corresponding kernel $K(X^i, X^j)$?

- [3 points] Suppose we want to train a hard-margin linear SVM in this **mapped** feature space. What is the Lagrangian dual problem?

- [3 points] Suppose we use SMO algorithm to solve the above optimization problem. We fix $\alpha_1 = \alpha_4 = 1/4$ and update α_2 and α_3. What are the updated α_2 and α_3, respectively?

- [6 points] Suppose we use the above α_i $(i = 1, 2, 3, 4)$ as the final solution for the Lagrangian dual problem. What is the weighted vector w? What is the off-set scalar b? What is the decision boundary of your SVM classifier?

6 Handwritten Digits Recognition with Logistic Regression and kNN [40 points]

In this question, you need to implement logistic regression model and k-Nearest Neighbor (kNN) algorithm for MNIST handwritten digits recognition. The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 images as training set and 10,000 images as test set. Each image has 28×28 pixels which are considered as the features of the image. You can download the data set from Blackboard. To ease your work, two additional code files are provided to load the images and labels. You need to implement both models by your own and use it to predict the digits in the test set. Since the data set is large, you might need to come up with a good way to optimize your algorithm and speedup the computation. If you are using Python, the only third party libraries allowed are *numpy*, *scipy* and *matplotlib*.

Note: Due to the large size of this dataset, you may want to start this problem earlier.

- [20 points] **Logistic Regression**. Consider a logistic regression model for multi-class classification. Assume that we have K different classes and each input $\mathbf{x}$ is a d dimensional vector. The posterior probability are shown as follows:

$$P(Y = k|X = \mathbf{x}) = \frac{exp(\mathbf{w}_k^T \mathbf{x})}{1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T \mathbf{x})}, \qquad \text{for } k = 1, ..., K-1$$
$$P(Y = K|X = \mathbf{x}) = \frac{1}{1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T \mathbf{x})},$$

  where $\mathbf{w}_k^T$ denotes the transpose of the $\mathbf{w}_k$ weight vector and $\mathbf{w}_k^T \mathbf{x}$ is the inner product of $\mathbf{w}_k^T$ and $\mathbf{x}$. To simplify, we **IGNORE** the constant term.

  Here you need to implement the above multi-class logistic regression model and use the **gradient ascent algorithm** to train your classifier. Please **DO NOT** add regularization term. Save your results and plot the testing accuracy vs. the number of iterations. Write your observations and explanations of the result from the figure into your report and submit it along with both the code and results (including the accuracy and figure). For hyper-parameter settings, you can set the tolerance to be $1e-4$ and maximum number of iterations to be 100.

- [20 points] **kNN**. For kNN, use Euclidean distance to measure the distance between each two data points. Choose the number of neighbors, $k = 1, 3, 5, 10, 30, 50, 70, 80, 90, 100$ and compute the prediction accuracy. Save your results and plot the accuracy vs. the number of neighbors. Write your observations and explanations of the result from the figure into your report and submit it along with both the code and results (including the accuracy and figure).