CS 475 Machine Learning: Project 1
Supervised Classifiers 1
Due: Thursday February 22, 2018, 11:59pm
50 Points Total Version 1.0

Junhao Xiong

1 Analytical (20 Points)

1) Supervised vs. Unsupervised Learning (3 points) .

1. Give an example of a problem that could be solved with both supervised learning and unsupervised learning. Is data readily available for this problem? How would you measure your 'success' in solving this problem for each approach?

   Answer: Recognizing hand written digits. One can either do classification i.e. supervised learning, or clustering i.e. unsupervised learning. The data is available - MNIST. Success in classification can be measured by the proportion of digits classified correctly. Success in clustering can be measured by the proportion of digits in the correct cluster. Both should be measured on a test dataset and compare the results to the ground truth.

2. What are the pros and cons of each approach? Which approach do you think the problem better lends itself to?

   Answer:
   Supervised learning
   Pro: It is relatively easier to get good accuracy. There exist many algorithms with guarantee for good performance. It is also generally more efficient.
   Con: Someone needs to hand label the training set.

   Unsupervised learning
   Pro: No need to hand label data
   Con: It is relatively harder to get good performance. It is generally slower. Many good clustering algorithms involved pairwise calculations of the entire training sets, which can be quite slow.

   Supervised learning is more natural for this problem, since there is already abundant datasets available for training.

2) Model Complexity (3 points) Explain when you would want to use a simple model over a complex model and vice versa. Are there any approaches you could use to mitigate the disadvantages of using a complex model?

Answer: One would want a simple model when the number of data available for training is relatively small and when one wants to prevent overfitting. A complex model is suitable when large number of data is available for training to achieve good performance and when one wishes to minimize bias.

To mitigate the problem of overfitting, one could perform regularization.

3) Training and Generalization (3 points) Suppose you're building a system to classify images of food into two categories: either the image contains a hot dog or it does not. You're given a dataset of 25,000 (image, label) pairs for training and a separate dataset of 5,000 (image, label) pairs.

1. Suppose you train an algorithm and obtain 96% accuracy on the larger training set. Do you expect the trained model to obtain similar performance if used on newly acquired data? Why or why not?

   Answer: One would not necessarily know the models performance on the test set, since despite having good performance on the training set, the model might be overfitting, or it might generalize well. One would need to know more about the model to answer the question.

2. Suppose that, after training, someone gives you a new test set of 1,000 (image, label) pairs. Which do you expect to give greater accuracy on the test set: The model after trained on the dataset of 25,000 pairs or the model after trained on the dataset of 5,000 pairs? Explain your reasoning.

   Answer: It depends. More data does not necessarily lead to better performance. The data might be poorly labeled, contain lots of noises, or the model might be poor. But given good data and a sound model, one may expect better performance for the model trained on more data.

3. Suppose your models obtained greater than 90% accuracy on the test set. How might you proceed in hope of improving accuracy further?

   Answer: One may tune the hyper-parameters using a validation set. But one should note to reinitialize all parameters before validation since the model has already seen the test set at this point.

4) Loss Function (3 points) State whether each of the following is a valid loss function for binary classification. Wherever a loss function is not valid, state why. Here y is the correct label and $\hat{y}$ is a decision confidence value, meaning that the predicted label is given by $\text{sign}(\hat{y})$ and the confidence on the classification increases with $|\hat{y}|$.

1. $\ell(y, \hat{y}) = \frac{3}{4}(y - \hat{y})^2$. Yes
2. $\ell(y, \hat{y}) = |(y - \hat{y})|/\hat{y}$. No, because it would be negative when $\hat{y}$ is negative.
3. $\ell(y, \hat{y}) = \max(0, 1 - y \cdot \hat{y})$. Yes

5) Linear Regression (4 points) Suppose you observe n data points $(x_1, y_1), \ldots, (x_n, y_n)$, where all x_i and all y_i are scalars.

1. Suppose you choose the model $\hat{y} = wx$ and aim to minimize the sum of squares error $\sum_i (y_i - \hat{y}_i)^2$. Derive the closed-form solution for w from scratch, where 'from scratch' means without using the least-squares solution presented in class.

$$\min \sum_i (y_i - \hat{y}_i)^2 = \min \sum_i y_i^2 - 2wx_i y_i + w^2 x_i^2$$

$$\partial((y_i - \hat{y}_i)^2)/\partial(w) = \sum_i (-2x_i y_i + 2wx_i^2) = 0$$

$$2w \sum_i x_i^2 = 2 \sum_i x_i y_i$$

$$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

2. Suppose you instead choose the model $\hat{y} = w \sin(x)$ and aim to minimize the sum of squares error $\sum_i (y_i - \hat{y}_i)^2$. Is there a closed-form solution for w? If so, what is it?

$$\min \sum_i (y_i - \hat{y}_i)^2 = \min \sum_i (y_i - wsin(x_i))^2 = \min \sum_i y_i^2 - 2wy_i sin(x_i) + w^2 sin(x_i)^2$$

$$\partial((y_i - \hat{y}_i)^2)/\partial(w) = \sum_i (-2y_i sin(x_i) + 2wsin(x_i)^2) = 0$$

$$w = \frac{\sum_i y_i sin(x_i)}{\sum_i sin(x_i)^2}$$

6) Logistic Regression (4 points) Explain whether each statement is true or false. If false, explain why.

1. In the case of binary classification, optimizing the logistic loss is equivalent to minimizing the sum-of-squares error between our predicted probabilities for class 1, $\hat{\mathbf{y}}$, and the observed probabilities for class 1, $\mathbf{y}$.

   Answer: False. First, to clarify, given $\hat{y}$ and y, we optimizing for maximum log likelihood for logistic regression or logistic loss, which are two different things. The negative log likelihood loss function is $\sum_i -logp(y_i = 1|x_i, w) = \sum_i -log\frac{1}{1+e^{-w^\intercal x_i}}$ for logistic regression. The logistic loss is $\sum_i log(1 + e^{-y_i w^\intercal x})$. The sum of square error can be expressed as $\sum_i (y_i - \hat{y}_i)^2$. These are different loss functions, so one will get different gradients when performing gradient descent on each of the three.

2. One possible advantage of stochastic gradient descent is that it can sometimes escape local minima. However, in the case of logistic regression, the global minimum is the only minimum, and stochastic gradient descent is therefore never useful.

   Answer: False. Despite that stochastic gradient is different from the gradient average over the entire dataset or batch, SGD is much more efficient, since it only calculates gradient for one sample for updates, instead of the entire sample set.