Wanze Xie
A92061040
Cogs 185 - HW1

* Code of this assignment is attached to a zip file along with the submission

1. Task 1: One-vs-All SVM

(1) Mathematic form of the gradient of the loss function:

$$L(w) = \frac{1}{2}\|w\|^2 + \lambda \sum \max(0, 1 - y_i \times f(x_i; w))$$

$$\frac{dL(w)}{dw} = w + \frac{\partial}{\partial w} \lambda \sum \max(0, 1 - y_i \times f(x_i; w))$$

for i, when $(1 - y_i \times f(x_i; w)) > 0$

$$\frac{\partial}{\partial w} \lambda \cdot \max(0, 1 - y_i \times f(x_i; w))$$

$$= \lambda \cdot \frac{\partial}{\partial w} (1 - y_i \times (x_i \cdot w))$$

$$= \lambda \cdot (0 - y_i \times x_i)$$

$$= -\lambda y_i \cdot x_i$$

when $(1 - y_i \times f(x; w)) \leq 0$

$$\frac{\partial}{\partial w} \lambda \cdot \max(0, 1 - y_i \times f(x_i; w))$$

$$= \lambda \cdot \frac{\partial}{\partial w}(0)$$

$$= 0$$

(2) After training, the optimal w* reported from the training program is:

w0 = (0.0934, 0.1040, 0.5980, -0.8146, -0.390)
w1 = (1.233, 0.8995, -1.977, -0.0850, -0.7523)
w2 = (-1.912, -1.219, -1.332, 2.033, 2.085)

And the optimal w* is optimal across the training with all different lamda values.

Note that w0 means the w* for target with label 0, w1 means the w* for the target with label 1, and w2 means the w* for the target with label 2.

(3) After training, the training accuracy and test accuracy resulted from different lamda values are:

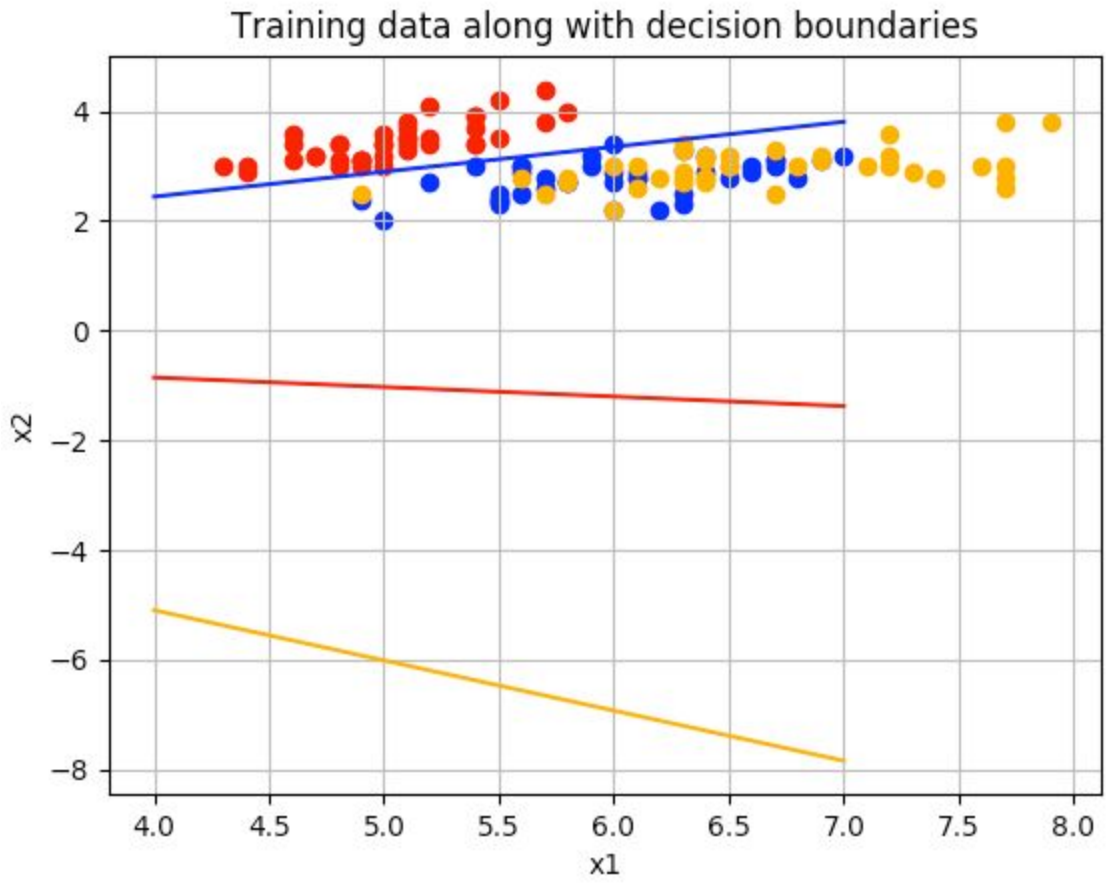
Learning rate is 0.00001
When lamda = 0.5,
Training accuracy = 88.33%
Testing accuracy = 86.67%

When lamda = 2.0
Training accuracy = 94.27%
Testing accuracy = 100%

When lamda = 5.0
Training accuracy = 94.27%
Testing accuracy = 100%

When lamda = 10.0
Training accuracy = 94.27%
Testing accuracy = 100%

(4) Plot training data along with decision boundaries with the best training result, which is learning_rate = 0.00001, lamda = 0.5



Note: the red dots are the targets with label 0, and blue dots are the targets with label 1, and orange dots are the targets with label 2. No specific meanings for the color of the decision boundary. Please also note that the decision boundary is drawn using the first two features of the data and the weights including bias, so the real decision boundary will be a high dimensional hyperplane.

2. Task 2: Explicit Multiclass SVM

(1) Mathematic form of the gradient of the loss function:

$$L(w_1, \cdots, w_k) = \frac{1}{2}\sum_{k=1}^{k}\|w_k\|^2 + \lambda \sum_{i}\sum_{m=1, m\neq y_i}^{k} \max(0, 1-(\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle))$$

$$\text{gradient} = \frac{\partial}{\partial w_j} L = \sum_{k=j} w_k + \lambda \sum_{i}\sum_{m=1, m\neq y_i}^{k} \frac{\partial}{\partial w_j} \max(0, 1-(\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle))$$

① $y_i = j$: $\frac{\partial}{\partial w_j} \max(0, 1-(\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle)) = \begin{cases} -x_i & \text{— (1)} \\ 0 & \text{— (2)} \end{cases}$

(1) case if $\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle < 1$

(2) Case if otherwise.

② $y_i \neq j$: $\frac{\partial}{\partial w_j} \max(0, 1-(\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle)) = \begin{cases} x_i & \text{— (3)} \\ 0 & \text{— (4)} \end{cases}$

(3) case if $\langle w_{y_i}, x_i\rangle - \langle w_k, x_i\rangle < 1$, and $k = j$

(4) case if otherwise

(2) After training, the optimal w* reported from the training program is:

w_1 *** = (0.2029, 0.4585, 0.8953, -1.209, -0.6516)**

w_2 *** = (0.4572, 0.4368, 0.09444, -0.4022, -0.4832)**

w_3 *** = (-0.6602, -0.8953, -0.9897, 1.611, 1.135)**

(3) After training, the training accuracy and test accuracy resulted from different lamda values are:

Learning rate is 0.000005
When lamda = 0.5,
Training accuracy = 96.67%
Testing accuracy = 100%

When lamda = 2.0
Training accuracy = 97.5%
Testing accuracy = 100%

When lamda = 5.0
Training accuracy = 95.83%
Testing accuracy = 96.66%

When lamda = 10.0
Training accuracy = 93.33%
Testing accuracy = 93.33%

(4) Plot training data along with decision boundaries with the best training result, which is learning_rate = 0.00001, lamda = 0.5

Note: the red dots are the targets with label 0, and blue dots are the targets with label 1, and orange dots are the targets with label 2. No specific meanings for the color of the decision boundary. Please also note that the decision boundary is drawn using the first two features of the data and the weights including bias, so the real decision boundary will be a high dimensional hyperplane.

3. Off-the-shelf Classifiers

* Random Forest is chosen as classifier for this problem

3.1. One-vs-All

By converting labels and obtaining 3 new datasets, in which in the label $y_i \in \{+1, -1\}$, we get the training result using the output of 3 classifiers on 3 bi-class datasets as following:

```
--------- Running Random Forest on iris data One-vs-All --------------
Training Error is  0.0416666666667
Test Error is  0.0


--------- Running Random Forest on DNA data One-vs-All  --------------
Training Error is  0.075
Test Error is  0.101180438449
```

3.2. Explicit Multiclass

Without any processing of labels, I directly trained and tested the random forest classifier on the dataset. The description of the classifier is explained in the report in the section 5. Here we will only list the training result as following:

```
--------- Running Random Forest on iris data Explicit Multiclass --------------
Training Error is  0.0583333333333
Test Error is  0.0333333333333


--------- Running Random Forest on DNA data Explicit Multiclass  --------------
Training Error is  0.108
Test Error is  0.150084317032
```

4. Bonus Points

This part is examined by referring to the paper published by Jorg Kindermann and Edda Leopold.

5. Report Requirement

* This report summarizes and compares the work and researches done in the above sections

5.1 Abstract

There are are diverse amount of multiclass classifiers created in order to complete the classification task that cannot be handled by the binary classifiers. Designing of a multiclass classifiers requires both high training and testing accuracy and favorable amount of training time. Intuitively, sophisticated multiclass classifiers seems more powerful as it can take in multi labels directly to process training. Nevertheless, this report hypothesize that in many cases, if not all, simple multiclass classifier evolved from binary classifier, which we call one-vs-all multiclass classifier, will have better performance in terms of both training time and accuracy.

5.2 Introduction

This report primarily address the comparison between the one-vs-all multiclass classifier and the explicit multiclass classifier. By approaching the multiclass training using both SVM and Random Forest, we seeks to substantiate our hypothesis that one-vs-all multiclass classifier will have better performance in some cases.

Similar hypothesis and experiment was also performed in Chih-Wei Hsh and Chih-Jen Lin's work [1] and also Ryan Rifkin and Aldebaro Klautau's work [2]. In the experiment of this report, we will use the iris dataset and DNA datasets from LIBSVM [3]

5.3 Method and Experiment

5.3.1 One-vs-All SVM

In this portion, we implemented 3 linear SVM using gradient descent method and iris data are used in this experiment. For each SVM, we converted the labels from

3 classes to 2 classes. In other words, we took turns to convert label 0, 1, and 2 into +1 and rendered the other 2 labels to -1. During prediction, phase, we predicted the example x with the 3 SVMs and took the results as degrees of belief to decide the predicted class.

The loss function we use, is as shown below, where $f(x_i; w) = < w, x_i >$

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum max(0, 1 - y_i \times f(\mathbf{x}_i; \mathbf{w})),$$

By tuning parameters, the best training result is obtained from the setting of learnin_rate to 0.00001, λ value to 2.0, with epochs of 50,000. The tuned weights are shown as following:

w0 = (0.0934, 0.1040, 0.5980, -0.8146, -0.390)
w1 = (1.233, 0.8995, -1.977, -0.0850, -0.7523)
w2 = (-1.912, -1.219, -1.332, 2.033, 2.085)

And the best training accuracy we obtained is 94.27%, where the best testing accuracy obtained is 100%. It is reasonable since we used 20% of the 150 data datasets as the testing sets so the test set is small enough to be predicted by the training model.

5.3.2 Explicit Multi-class SVM

Explicit Multiclass SVM differs from one-vs-all SVM in a way that explicit multi-class SVM utilizes multiple hyperplane [4] to classify data. Similarly, in this experiment we also used the gradient descent method to implement the multi-class SVM and iris data sets is used in the experiment.

Since the iris data sets have 3 classes, we set K=3 to refer to the total number of classes. The loss function we based on is as following:

$$minimize \qquad L(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2$$

$$+ \quad \lambda \sum_i \sum_{k=1, k\neq y_i}^{K} max\left(0, 1 - (< \mathbf{w}_{y_i}, \mathbf{x}_i > - < \mathbf{w}_k, \mathbf{x}_i >)\right)$$

By tuning the parameters, we obtained the the best w* = ($w_1, w_2 \ldots w_k$) to be like following:

w_1^* = (0.2029, 0.4585, 0.8953, -1.209, -0.6516)

w_2^* = (0.4572, 0.4368, 0.09444, -0.4022, -0.4832)

w_3^* = (-0.6602, -0.8953, -0.9897, 1.611, 1.135)

The best training accuracy we obtained is 97.5%, and the best training result is still 100%. We uses the same datasets for the training data and testing data. And so far it looks like the performance of this two models are similar to each other and there is no significant difference in the difference of the two.

5.3.3 One-vs-All Random Forest

The SVM classifier does not show significant performance, we believe the reasons are due that the SVM classifier might be less sensitive to the One-vs-All and Explicit Multiclass model. On the other hand, the iris dataset is too limited to show the difference, so in this experiment, we adopted Random Forest Classifier and will implement the One-vs-All and Explicit Multiclass Model based on the top of this classifier and compare the performance difference. The datasets we use in this example will be both the iris dataset and the DNA dataset.

The random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting [5]. Conceptually, this classifier can take any number of labels and itself can be considered as a multiclass classifier on its own. But here we design it specifically so that it can only take two labels that is {+1, -1} and we created a wrapper methods to make it a one-vs-all multiclass classifier that classifies these two 3-class datasets.

The training accuracy we obtained using iris dataset is 95.93% and testing accuracy is 100%. For dna datasets, the training accuracy we obtained is 92.50%, and the testing accuracy is 90.89%. Note that the max_depth we used for iris dataset is 2, and for dna datasets is 5 so that we can obtain a reasonable training result.

5.3.4 Explicit Multiclass Random Forest

In this last experiment we uses the same setting of random forest to keep the validity of the experiment. Specifically, we train the random forest by setting the max_depth to 2 for the iris dataset, and max_depth to 5 for the dna dataset. Please note that since the random forest classifier from sklearn already has the ability to achieve multiclass classification tasks, we simply add a wrapper for this classifier and parsing the training dataset to the classifier to obtain the result.

The result we obtained is a training accuracy of only 94.17%, and testing accuracy of only 96.67% for the iris dataset. And a training accuracy of 90.20% and a testing accuracy of only 84.99% for the dna datasets.

In this pair of experiment, it became clear that when the datasets get larger, and when different core classifiers are adopted, one-vs-all model will show a better performance and yield better training result.

5.4 Conclusion

In this study and report, we have been discussed the situations under which the one-vs-all multiclass classification model would result to a better performance than the explicit multiclass model. The two core classifiers we experimented is the SVM classifier with different loss functions based on the gradient descent method, and the random forest classifier provided by the sklearn. We configured the two classifiers in the same way so that the training result is comparable. For SVM, we see that no significant difference in the performance, but in random forest classifier, we do see that one-vs-all multiclass classifier achieves a better performance than the explicit multiclass classifier. A side discovery we made is that at least on the iris and dna datasets, within a range from 1 to 5, the deeper the max_depth of the random forest, the better the training result will be.

As a summary of the paper, we can conclude that one-vs-all classifier conceptually is easier to configure and set-up and less calculation is required during the training. In the real world use, we see that the one-vs-all classifier can performs better than explicit multiclass classifiers. It is still too hasty to conclude when will be the best case to adopt one-vs-all model in the multiclass classification

task, but we believe it is worthwhile to try out this classification model to see if a favorable performance and training result is achieved.

5.5 Acknowledgement

This report adopted some wordings from the assignment write up to makes the explanation of certain terms easier for understanding and interpreting. The help from professor Tu and TA is deeply appreciated and otherwise this report will be less enriched and sophisticated.

5.6 Reference

[1] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. Neural Networks, IEEE Transactions on, 13(2):415–425, 2002. 4

[2] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. The Journal of Machine Learning Research, 5:101–141, 2004.

[3] LIBSVM Data: Classification (Multiclass)
https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html

[4] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. Journal of machine learning research, 2(Dec):265–292, 2001.

[5] sklearn.ensemble.RandomForestClassifier
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html