CSE 6363: Machine Learning
University of Texas at Arlington
Summer 2022
Alex Dillhoff

Assignment 1

This assignment covers linear regression and logistic regression using disciminative methods.

1 The Iris Dataset

The Iris flower data set (https://en.wikipedia.org/wiki/Iris_flower_data_set) was organized by Ronald Fisher in 1936. It is a commonly used dataset for introductory machine learning concepts. You will use this dataset for use with a classification AND regression task.

1.1 Preparing the Data

To begin, load the data using `scikit-learn`. As we saw during class, the setosa samples are very clearly linearly separable given any combination of two features. However, the versicolor and virginica usually have some overlap.

In order to verify the models that you will create in the following two sections, you will need to take some portion of the dataset and reserve it for testing. Randomly select 10% of the dataset, ensuring an even split of each class. This will be your **test** set. The rest of the data will serve as your **training** set.

2 Regression

Fit 12 linear regression models to the training data with parameters $\mathbf{w} = [w_0, w_1]$ for each one. For example, your first model may use sepal length as the input feature to predict sepal width and the second model would reverse that combination. Some of these input features will not be good predictors.

2.1 Model Definition

Your implementation should define a class for `LinearRegression` which includes at least a `fit` and `predict` method. Additional methods can be added as you see fit.

The `fit` method should accept 2 required parameters: the input data and target values. Other parameters can be added as long as they are optional.

2.2 Training

Your models should be trained using batch gradient descent with a batch size (optional parameter) of 32. Use mean squared error as your loss function. For each model, train for $n = 100$ steps (optional parameter). As each model trains, record the loss average over the batch size against the current step number. One way to save this data is to either return an array from the `fit` method or save it as an internal class member that can be retrieved after training is complete. **Plot the loss against the step number and save it. This will go in your report.**

To observe the effects of regularization, pick **one** of your trained models and inspect the weights. Train an identical model again, except this time you will add L2 regularization to the loss. Record the difference in parameters between the regularized and non-regularized model. **In your report, include the weight values in this comparison**.

2.3 Testing

For each model you created, test its performance on unseen data by evaluating the mean squared error against the test dataset that you set aside previously. Based on these results, which input feature is most predictive of its corresponding output feature? **Create a table of results that summarized the testing accuracy of each model and put it in your report.**

3 Classification

For classification, you will implement Linear Discriminant Analysis, Logistic Regression, and Naive Bayes models for classification. Your implementation should define a class for each model which includes at least a `fit` and `predict` method. Additional methods can be added as you see fit.

3.1 Training

The models can be fit following the parameter fitting methods discussed in class for each respective model.

3.2 Testing

For each model you created, test its performance on unseen data by evaluating the mean squared error against the test dataset that you set aside previously. Based on these results, which input feature is most predictive of its corresponding output feature? **Create a table of results that summarized the testing accuracy of each model and put it in your report.**

Submission

Create a zip file that includes all of your code as well as your report. The TA should be able to easily run the code to reproduce all plots and results. Include any additional instructions, if necessary.