README

OMSCS CS7641-A1

Assignment Requried Materials

- Github Repo (public repo): https://github.com/HaoLiangPao/CS7641-ML
- Overleaf Link (read only, public accessible): https://www.overleaf.com/read/wvzftzbvzrbg#1d14ec

Overview

This repository contains scripts for analyzing various machine learning models on different datasets. The primary focus is on SVM, KNN, NN, and Boosting models. The analysis includes data loading, preprocessing, training, hyperparameter tuning, and plotting validation and learning curves.

Repository Structure

- **analysis_configs.json5**: Configuration file containing hyperparameters for different models and datasets.
- **plotting.py**: Utility functions for plotting learning and validation curves.
- **NN.py**: Script for Neural Network analysis.
- **SVM.py**: Script for Support Vector Machine analysis.
- **KNN.py**: Script for K-Nearest Neighbors analysis.
- **Boosting.py**: Script for Boosting (Decision Tree) analysis.
- **dir(wine)** & **dir(adult_census_income)**: Similar experinment-framework but used seperatly as they are for two datasets.

Setup Instructions

1. **Clone the repository:**

```shell
git clone https://github.com/HaoLiangPao/CS7641-ML.git
cd <CS7641-ML>
```

2. **Install required packages:**

   ```shell
   ./start.sh
   ```

   This script will create python virtual environment and install the dependencies for you to run the analysis.

Using the Scripts

1. Analysis Files

These files are used to load data, process it, train the models, and plot validation curves for hyperparameter tuning.

- **Load Data**: Scripts in the `wine`/`adult_census_income` directories handle loading and preprocessing of datasets. (in this case, both datasets are downloaded from the internet through URLs, so no extra dataset files needed)
- **Process Data**: Data is scaled and split into training and test sets.
- **Train Model**: Models are trained using different hyperparameters.
- **Plot Validation Curve**: Validation curves are plotted to visualize the effect of different hyperparameters.

2. Configuration Files

The hyperparameter analysis depends on the range of hyperparameters been defined inside the `analysis_configs.json5`.

After analyzing the validation curves, update the `best_configs.json5` file with the best hyperparameters. This file should reflect the optimal settings determined from the validation curve analysis.

3. Individual Analysis Files

Run the individual analysis scripts to train the models with the tuned hyperparameters and plot the final learning curves.

The images with the corresponding **dataset_name**, **algorithm_name**, **purpose** and **plot_name** will be generated within the `images` folder inside.

Example: Running SVM Analysis

1. **Update Configuration File:**
   Ensure
   1. the `best_configs.json5` file contains the best hyperparameters for the SVM model.
   2. the best hyperparameters have selected from the plots generated previously.
2. **Run SVM Analysis Script:**

   ```shell
   python SVM.py
   ```

   This script will:

   - Load and preprocess the data.
   - Train the SVM model with the optimal hyperparameters.
   - Plot the final learning curves using functions defined in `plotting.py`.

Utility Functions

The `plotting.py` file contains utility functions used across different analysis scripts:

- **plot_learning_curve**: Plots the learning curve of a model.
- **plot_validation_curve**: Plots the validation curve for hyperparameter tuning.
- **plot_learning_curve_with_test**: Plots the learning curve including the test set performance.
- **plot_iterative_learning_curves**: Plots iterative learning curves for multiple models.
- **plot_multiple_learning_curves**: Plots learning curves for different kernel functions in a single graph.

Example Workflow

1. **Hyperparameter Tuning:**
   Run the main analysis script to tune hyperparameters and plot validation curves.

   ```shell
   python main_analysis.py
   ```

2. **Update Configuration:**
   Edit `best_configs.json5` with the best hyperparameters found in the previous step.

3. **Final Analysis:**
   Run the individual analysis scripts to train the models with the tuned hyperparameters and plot the final learning curves.

SHELL

```
cd wine
python NN.py
python SVM.py
python KNN.py
python Boosting.py
ls -la images

cd wine
python NN.py
python SVM.py
python KNN.py
python Boosting.py
ls -la images
```

Notes

- Customize the configuration file based on the specific requirements of your dataset and model.

Conclusion

This repository provides a framework for comprehensive analysis of machine learning models, including hyperparameter tuning and visualization of model performance. By following the steps outlined above, you can apply these methods to your own datasets and models.