XLearning

Release 0.0.2

Rongting Huang

Sep 07, 2022

MAIN

CHAPTER

ONE

ABOUT XLEARNING

A small library of machine learning models and utility & plotting functions:

1. a set of utility functions, e.g., wrap function for cross-validation on regression and classification models
2. a set of small models, e.g., mixture of linear regression model
3. a set of plotting functions, e.g., *corr_plot*, *ROC_curve*, *PR_curve*

1.1 Installation

1.1.1 Easy install

xlearning is available through pypi. To install, type the following command line, and add `-U` for upgrading:

```
pip install -U xlearning
```

Alternatively, you can install from this GitHub repository for latest (often development) version by following command line

```
pip install -U git+https://github.com/Rongtingting/XLearning
```

In either case, if you don't have write permission for your current Python environment, add `--user`, but check the previous section on create your own conda environment.

To remove the package, type the following command line,

```
pip uninstall xlearning
```

1.1.2 Issues

If you have any issue, please report it to the issue on XLearning github.

{

"cells": [

{

"cell_type": "markdown", "metadata": {}, "source": [

"# Cross validation"

]

}, {

“cell_type”: “code”, “execution_count”: 1, “metadata”: {}, “outputs”: [], “source”: [

“import h5pyn”, “import numpy as npn”, “import scipy.stats as stn”, “import matplotlib.pyplot as pltn”, “n”, “# regressionn”, “f = h5py.File("../data/regression_data.hdf5", "r")n”, “X = np.array(f["X"])n”, “Y = np.array(f["Y"])n”, “f.close()”

]

}, {

“cell_type”: “code”, “execution_count”: 2, “metadata”: {}, “outputs”: [

{

“name”: “stdout”, “output_type”: “stream”, “text”: [

“0.2.2n”

]

}

], “source”: [

“import hilearnn”, “print(hilearn.__version__)”

]

}, {

“cell_type”: “markdown”, “metadata”: {}, “source”: [

“## Regression”

]

}, {

“cell_type”: “code”, “execution_count”: 3, “metadata”: {}, “outputs”: [

{

“data”: {

“image/png”: “iVBORw0KGgoAAAANSUhEUgAAAx4AAAAEQCAYAAAAA+r6JnAAAAOXRFWHRTb2Z0

“text/plain”: [

“<Figure size 800x280 with 3 Axes>”

]

}, “metadata”: {

“needs_background”: “light”

}, “output_type”: “display_data”

}

], “source”: [

“from sklearn import linear_modeln”, “from sklearn.ensemble import RandomForestRegressorn”, “n”, “from hilearn import CrossValidation, corr_plotn”, “n”, “# define the model objectsn”, “linreg = linear_model.LinearRegression()n”, “lassoreg = linear_model.LassoCV(cv=3)n”, “randforest = RandomForestRegressor(n_estimators=100, n_jobs=-1)n”, “n”, “fig = plt.figure(figsize=(10, 3.5), dpi=80)n”, “plt.subplot(1,3,1)n”, “# Cross-Validation wrap & corr_plotn”,

```
"CV = CrossValidation(X,Y)\n", "Y_pre = CV.cv_regression(linreg)\n", "corr_plot(Y, Y_pre, size=20)\n", "\n", "plt.xlabel("observation")\n", "plt.ylabel("prediction")\n", "plt.title("linear regression")\n", "plt.grid(alpha=0.2)\n", "# pl.ylim(-6,4)\n", "\n", "plt.subplot(1,3,2)\n", "CV = CrossValidation(X,Y)\n", "Y_pre = CV.cv_regression(lassoreg)\n", "corr_plot(Y, Y_pre, size=20)\n", "plt.xlabel("observation")\n", "plt.ylabel("prediction")\n", "plt.title("Lasso regression")\n", "plt.grid(alpha=0.2)\n", "# pl.ylim(-6,4)\n", "\n", "plt.subplot(1,3,3)\n", "CV = CrossValidation(X,Y)\n", "Y_pre = CV.cv_regression(randforest)\n", "corr_plot(Y, Y_pre, size=20)\n", "plt.xlabel("observation")\n", "plt.ylabel("prediction")\n", "plt.title("random forest regression")\n", "plt.grid(alpha=0.2)\n", "# pl.ylim(-6,4)\n", "\n", "plt.tight_layout()\n", "# fig.savefig("cv_regression.pdf", dpi=300, bbox_inches='tight')\n", "plt.show()"
    ]
}, {
    "cell_type": "markdown", "metadata": {}, "source": [
        "## Classification"
    ]
}, {
    "cell_type": "code", "execution_count": 4, "metadata": {}, "outputs": [], "source": [
        "# classificationn", "# making pseudo-label by stratifying y valuesn", "n", "idx = (Y < -0.5) + (Y > 0.5)n", "X1 = X[idx,:]n", "Y1 = (Y[idx] > 0.5).astype("int")"
    ]
}, {
    "cell_type": "code", "execution_count": 5, "metadata": {}, "outputs": [
        {
            "data": {
                "image/png": "iVBORw0KGgoAAAANSUhEUgAAAU4AAAEUCAYAAABERt/4AAAAOXRFWHRTb
                    "<Figure size 360x280 with 1 Axes>"
                ]
            }, "metadata": {
                "needs_background": "light"
            }, "output_type": "display_data"
        }
    ], "source": [
        "from sklearn import linear_modeln", "from sklearn.ensemble import RandomForestClassifiern", "n", "from hilearn import ROC_plot, CrossValidationn", "n", "LogisticRegression = linear_model.LogisticRegression(solver='lbfgs')n", "RF_class = RandomForestClassifier(n_estimators=100, n_jobs=-1)n", "n", "CV = CrossValidation(X1, Y1)n", "Y1_pre, Y1_score = CV.cv_classification(model=RF_class, folds=10)n", "Y2_pre, Y2_score = CV.cv_classification(model=LogisticRegression, folds=10)n", "n", "fig = plt.figure(figsize=(4.5, 3.5), dpi=80)n", "ROC_plot(Y1, Y1_score[:,1], legend_label="Random Forest", threshold=0.5, base_line=False)n", "ROC_plot(Y1, Y2_score[:,1], legend_label="Logistic Regress", threshold=0.5)n",
```

“plt.title("ROC curve for classification")n”, “n”, “# fig.savefig("cv_classification.pdf", dpi=300, bbox_inches=’tight’)n”, “plt.show()”

]

}, {

“cell_type”: “code”, “execution_count”: 6, “metadata”: {}, “outputs”: [

{

“data”: {

“image/png”: “iVBORw0KGgoAAAANSUhEUgAAAU4AAAEUCAYAAABERt/4AAAAOXRFWHRTb

“text/plain”: [

“<Figure size 360x280 with 1 Axes>”

]

}, “metadata”: {

“needs_background”: “light”

}, “output_type”: “display_data”

}

], “source”: [

“from hilearn import PR_curven”, “n”, “fig = plt.figure(figsize=(4.5, 3.5), dpi=80)n”, “PR_curve(Y1, Y1_score[:,1], legend_label="Random Forest", threshold=0.5, base_line=False)n”, “PR_curve(Y1, Y2_score[:,1], legend_label="Logistic Regress", threshold=0.5)n”, “plt.title("Precision-recall curve for classification")n”, “n”, “# fig.savefig("cv_classification_PRCurve.pdf", dpi=300, bbox_inches=’tight’)n”, “plt.show()”

]

}, {

“cell_type”: “code”, “execution_count”: null, “metadata”: {}, “outputs”: [], “source”: []

}

], “metadata”: {

“kernelspec”: {

“display_name”: “Python 3”, “language”: “python”, “name”: “python3”

}, “language_info”: {

“codemirror_mode”: {

“name”: “ipython”, “version”: 3

}, “file_extension”: “.py”, “mimetype”: “text/x-python”, “name”: “python”, “nbconvert_exporter”: “python”, “pygments_lexer”: “ipython3”, “version”: “3.7.4”

}

}, “nbformat”: 4, “nbformat_minor”: 4

}