Tutorial_5_TrAdaBoost 02 April 2025

Transfer Learning for Class AMAT 6000A: Advanced Materials Informatics Spring 2025, HKUST (GZ)

This tutorial introduces the concept and implementation of an instance transfer learning algorithm, **TrAdaBoost**, which is an extension of the AdaBoost algorithm. We will go step by step an example of utilizing TrAdaBoost for property classification, using Python and Jpyter Noebooks.

Code Example, Data, and Illustrations

The codes and examples are provided by Bin CAO on https://github.com/Bin-Cao/TrAdaboost/tree/main/TrAdaBoost.

Preparing for the Class

To run the code examples in this tutorial, ensure you have Python and Jupyter Notebook installed. Below is a comprehensive guide to help you get set up

Requirements

- python >= 3.7
- sklearn

Introduction of TrAdaBoost

TrAdaBoost (Transfer AdaBoost) is a transfer learning algorithm designed to improve the performance of machine learning models when the training data and test data come from different distributions. It is an extension of the AdaBoost algorithm, specifically tailored for transfer learning scenarios. TrAdaBoost aims to adapt the source data to the target domain by reweighting the samples in the source data based on their relevance to the target domain. The key idea is to iteratively reweight the training instances, giving more importance to instances that are harder to classify correctly, and to combine multiple weak learners to form a strong learner.

Initialization and Setup

In [238…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import copy
from sklearn import tree
```

Function Definition and Explanation

In [240…

```
def TrAdaBoost(trans_S, trans_A, label_S, label_A, test, N = 20):
    """Boosting for Transfer Learning.
```

```
Please feel free to open issues in the Github : https://github.com/Bin-Cao/TrAdaboost
or
contact Bin Cao (bcao@shu.edu.cn)
in case of any problems/comments/suggestions in using the code.

Parameters
----------
trans_S : feature matrix of same-distribution training data

trans_A : feature matrix of diff-distribution training data

label_S : label of same-distribution training data, 0 or 1

label_A : label of diff-distribution training data, 0 or 1

test : feature matrix of test data

N : int, default=20
the number of weak estimators

References
----------
.. [1] Dai, W., Yang, Q., et al. (2007).
Boosting for Transfer Learning.(2007), 193--200.
In Proceedings of the 24th international conference on Machine learning.

.. [2] GitHub: https://github.com/chenchiwei/tradaboost/blob/master/TrAdaboost.py
"""

trans_data = np.concatenate((trans_A, trans_S), axis=0)
trans_label = np.concatenate((label_A, label_S), axis=0)

row_A = trans_A.shape[0]
row_S = trans_S.shape[0]
row_T = test.shape[0]

if N > row_A:
    print('The maximum of iterations should be smaller than ', row_A)

test_data = np.concatenate((trans_data, test), axis=0)

# Initialize the weights
weights_A = np.ones([row_A, 1]) / row_A
weights_S = np.ones([row_S, 1]) / row_S
weights = np.concatenate((weights_A, weights_S), axis=0)

bata = 1 / (1 + np.sqrt(2 * np.log(row_A / N)))

# Save prediction labels and bata_t
bata_T = np.zeros([1, N])
result_label = np.ones([row_A + row_S + row_T, N])

# Save the prediction labels of test data
predict = np.zeros([row_T])
print ('params initial finished.')
print('='*60)

trans_data = np.asarray(trans_data, order='C')
trans_label = np.asarray(trans_label, order='C')
test_data = np.asarray(test_data, order='C')

error_rate_list = []
misclassify_list = []
for i in range(N):
    P = calculate_P(weights)
```

```
        result_label[:, i] = train_classify(trans_data, trans_label, test_data, P)
        error_rate,misclassify = calculate_error_rate(label_S, result_label[row_A:row_A + row_
        if error_rate > 0.5:
            error_rate = 1 - error_rate
            # for a binary classifier
            # reverse the prediction label 0 to 1; 1 to 0.
            pre_labels = copy.deepcopy(result_label[:, i])
            result_label[:, i] = np.invert(pre_labels.astype(np.int32)) + 2
        # Avoiding overfitting
        elif error_rate <= 1e-10:
            N = i
            break
        error_rate_list.append(error_rate)
        misclassify_list.append(misclassify)
        bata_T[0, i] = error_rate / (1 - error_rate)
        print ('Iter {}-th result :'.format(i))
        print ('error rate :', error_rate, '|| bata_T :', error_rate / (1 - error_rate))
        print('-'*60)
        # Changing the data weights of same-distribution training data
        for j in range(row_S):
            weights[row_A + j] = weights[row_A + j] * np.power(bata_T[0, i], (-np.abs(result_
        # Changing the data weights of diff-distribution training data
        for j in range(row_A):
            weights[j] = weights[j] * np.power(bata, np.abs(result_label[j, i] - label_A[j]))

    for i in range(row_T):
        left = np.sum(
            result_label[row_A + row_S + i, int(np.floor(N / 2)):N] * np.log(1 / bata_T[0, in
        right = 0.5 * np.sum(np.log(1 / bata_T[0, int(np.floor(N / 2)):N]))
        if left >= right:
            predict[i] = 1
        else:
            predict[i] = 0
    print("TrAdaBoost is done")
    print('='*60)
    print('The prediction labels of test data are :')
    print(predict)
    return predict,np.round(np.array(error_rate_list),3),np.round(np.array(misclassify_list),

def calculate_P(weights):
    total = np.sum(weights)
    return np.asarray(weights / total, order='C')

def train_classify(trans_data, trans_label, test_data, P):
    clf = tree.DecisionTreeClassifier(criterion="gini", max_depth = 3, class_weight = 'balanc
    clf.fit(trans_data, trans_label, sample_weight=P[:, 0])
    return clf.predict(test_data)

def calculate_error_rate(label_R, label_H, weight):
    total = np.sum(weight)
    misclassify = np.sum(np.abs(label_R - label_H))/len(label_H)
    return np.sum(weight[:, 0] / total * np.abs(label_R - label_H)) , misclassify,
```

1. Parameters:

- `trans_S` : Feature matrix of the source domain training data.
- `trans_A` : Feature matrix of the auxiliary domain training data (different distribution).
- `label_S` : Labels of the source domain training data.
- `label_A` : Labels of the auxiliary domain training data.
- `test` : Feature matrix of the test data.
- `N` : Number of weak estimators (default is 20).

2. Data Concatenation

- **Purpose**: Combines the source and auxiliary domain data into a single dataset for training.

```
trans_data = np.concatenate((trans_A, trans_S), axis=0)
trans_label = np.concatenate((label_A, label_S), axis=0)
```

3. Initialization of Weights

- Initializes weights for each instance in the auxiliary and source domains. The weights are normalized to ensure they sum up to 1.

```
row_A = trans_A.shape[0]
row_S = trans_S.shape[0]
row_T = test.shape[0]

weights_A = np.ones([row_A, 1]) / row_A
weights_S = np.ones([row_S, 1]) / row_S
weights = np.concatenate((weights_A, weights_S), axis=0)
```

4. Parameter Initialization

- **Beta Calculation**: `bata` is a parameter that controls the rate at which weights are updated for the auxiliary domain instances.

```
bata = 1 / (1 + np.sqrt(2 * np.log(row_A / N)))
```

5. Training Loop

```
for i in range(N):
    P = calculate_P(weights)
    result_label[:, i] = train_classify(trans_data, trans_label, test_data, P)
    error_rate, misclassify = calculate_error_rate(label_S, result_label[row_A:row_A
+ row_S, i], weights[row_A:row_A + row_S, :])
```

- **Iteration**: The algorithm iterates `N` times, training a weak learner in each iteration.
- **Weighted Sampling**: `calculate_P(weights)` computes the probability distribution for sampling instances based on their weights.
- **Training Weak Learner**: `train_classify` trains a weak learner (e.g., a decision tree) on the weighted data.
- **Error Calculation**: `calculate_error_rate` computes the error rate of the weak learner on the source domain data.

6. Error Handling and Weight Update

```
if error_rate > 0.5:
    error_rate = 1 - error_rate
    pre_labels = copy.deepcopy(result_label[:, i])
    result_label[:, i] = np.invert(pre_labels.astype(np.int32)) + 2
elif error_rate <= 1e-10:
    N = i
    break
```

- **Error Handling**: If the error rate is greater than 0.5, the prediction labels are inverted (since the weak learner is performing worse than random guessing).
- **Early Stopping**: If the error rate is extremely low (less than a threshold), the algorithm stops early to prevent overfitting.

```
bata_T[0, i] = error_rate / (1 - error_rate)
for j in range(row_S):
    weights[row_A + j] = weights[row_A + j] * np.power(bata_T[0, i], (-
np.abs(result_label[row_A + j, i] - label_S[j])))
for j in range(row_A):
    weights[j] = weights[j] * np.power(bata, np.abs(result_label[j, i] -
label_A[j]))
```

- **Beta Update**: Updates the bata_T array, which stores the beta values for each iteration.
- **Weight Update**: Adjusts the weights of the source and auxiliary domain instances based on their prediction accuracy.

Experiment

1. Load Data

In [244…

```
# same-distribution training data
S_train_data = pd.read_csv('Sdata.csv')
# diff-distribution training data
A_train_data = pd.read_csv('Adata.csv')
# test data
test_data = pd.read_csv('Tdata.csv')
```

2. Data Inspection

In [246…

```
print("Same-distribution training data: ")
print(S_train_data, "\n")
print("Diff-distribution training data: ")
print(A_train_data, "\n")
print("Test data: ")
print(test_data, "\n")
```

```
Same-distribution training data:
     Sn   Bi   In   Ti  UTS(MPa)
0  90.1  3.5  1.8  0.1         0
1  93.8  0.5  0.8  0.4         0
2  94.1  0.5  0.8  0.1         0
3  92.1  0.5  2.8  0.1         0
4  88.8  3.5  2.8  0.4         1
5  89.1  3.5  2.8  0.1         1
6  91.1  3.5  0.8  0.1         1

Diff-distribution training data:
      Sn   Bi   In   Ti  UTS(MPa)
0   92.7  3.0  2.5  0.3         0
1   92.6  2.5  3.3  0.1         0
2   92.9  3.0  2.5  0.1         0
3   92.4  2.5  3.4  0.2         0
4   92.5  2.5  3.4  0.1         0
5   91.9  3.0  3.5  0.1         0
6   91.5  3.0  3.5  0.5         0
7   91.7  3.0  3.5  0.3         0
8   92.5  3.0  2.5  0.5         0
9   90.9  4.0  3.5  0.1         1
10  91.7  4.0  2.5  0.3         1
11  91.5  4.0  2.5  0.5         1
12  90.7  4.0  3.5  0.3         1
13  90.9  5.0  2.5  0.1         1
14  90.5  4.0  3.5  0.5         1
15  89.7  5.0  3.5  0.3         1
16  89.5  5.0  3.5  0.5         1
17  89.9  5.0  3.5  0.1         1
18  90.7  5.0  2.5  0.3         1
19  90.5  5.0  2.5  0.5         1

Test data:
     Ti   Bi   In  Ti.1  UTS(MPa)
0  91.8  0.5  2.8   0.4         0
1  93.1  0.5  1.8   0.1         0
2  90.6  2.0  2.8   0.1         0
3  89.8  3.5  1.8   0.4         1
4  90.8  3.5  0.8   0.4         1
```

3. Transfer Learning and Prediction

In [248…

```
trans_S = S_train_data.iloc[:,:-1]
trans_A = A_train_data.iloc[:,:-1]
label_S = S_train_data.iloc[:, -1]
label_A = A_train_data.iloc[:,-1]
test = test_data.iloc[:,:-1]
pre, _, _ = TrAdaBoost(trans_S, trans_A, label_S, label_A, trans_S, 8)

print("Ground truth: ", np.array(label_S))
```

```
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
Ground truth:  [0 0 0 0 1 1 1]
```

4. Visualization

To illustrate how prediction errors change with iteration number, we test the first 20 iterations.

In [250…

```
"""
# example of book
# [an introduction of materials informatics II, Tong-yi Zhang]
"""

N_iter = 20
pre_err = []
for i in range(N_iter):
    N = i+1
    pre, _, _ = TrAdaBoost(trans_S, trans_A, label_S, label_A, trans_S, N)
    pre_err.append(sum(abs(pre - label_S))/len(trans_S))

_,error,misclassify_list = TrAdaBoost(trans_S, trans_A, label_S, label_A, test, N=20)
```

```
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 0.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[1. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[1. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 0. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
```

```
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 0. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 0. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 0. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
```

```
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
```

```
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
```

```
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
```

```
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953491 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
TrAdaBoost is done
```

```
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953488 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
```

```
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953488 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953488 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
```

```
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
------------------------------------------------------------
Iter 16-th result :
error rate : 0.1908711812165715 || bata_T : 0.23589714861912503
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953489 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
```

```
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
------------------------------------------------------------
Iter 16-th result :
error rate : 0.1908711812165715 || bata_T : 0.23589714861912503
------------------------------------------------------------
Iter 17-th result :
error rate : 0.19091367430396053 || bata_T : 0.23596205774423593
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
```

```
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
------------------------------------------------------------
Iter 16-th result :
error rate : 0.1908711812165715 || bata_T : 0.23589714861912503
------------------------------------------------------------
Iter 17-th result :
error rate : 0.19091367430396053 || bata_T : 0.23596205774423593
------------------------------------------------------------
Iter 18-th result :
error rate : 0.19094024787094088 || bata_T : 0.23600265291713904
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.1802325581395349 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
```

```
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
------------------------------------------------------------
Iter 16-th result :
error rate : 0.1908711812165715 || bata_T : 0.23589714861912503
------------------------------------------------------------
Iter 17-th result :
error rate : 0.19091367430396053 || bata_T : 0.23596205774423593
------------------------------------------------------------
Iter 18-th result :
error rate : 0.19094024787094088 || bata_T : 0.23600265291713904
------------------------------------------------------------
Iter 19-th result :
error rate : 0.1909565477846002 || bata_T : 0.2360275548386243
------------------------------------------------------------
TrAdaBoost is done
============================================================
The prediction labels of test data are :
[0. 0. 0. 0. 1. 1. 1.]
params initial finished.
============================================================
Iter 0-th result :
error rate : 0.14285714285714288 || bata_T : 0.1666666666666667
------------------------------------------------------------
Iter 1-th result :
error rate : 0.08333333333333334 || bata_T : 0.09090909090909093
------------------------------------------------------------
Iter 2-th result :
error rate : 0.2727272727272727 || bata_T : 0.37499999999999994
------------------------------------------------------------
Iter 3-th result :
error rate : 0.03125000000000001 || bata_T : 0.03225806451612904
------------------------------------------------------------
Iter 4-th result :
error rate : 0.17741935483870971 || bata_T : 0.215686274509804
------------------------------------------------------------
Iter 5-th result :
error rate : 0.15686274509803924 || bata_T : 0.186046511627907
------------------------------------------------------------
Iter 6-th result :
error rate : 0.18023255813953488 || bata_T : 0.2198581560283688
------------------------------------------------------------
Iter 7-th result :
error rate : 0.18085106382978722 || bata_T : 0.22077922077922074
------------------------------------------------------------
Iter 8-th result :
error rate : 0.18614718614718612 || bata_T : 0.22872340425531912
------------------------------------------------------------
Iter 9-th result :
error rate : 0.18749999999999994 || bata_T : 0.2307692307692307
------------------------------------------------------------
Iter 10-th result :
error rate : 0.18903436988543365 || bata_T : 0.2330978809283551
------------------------------------------------------------
Iter 11-th result :
error rate : 0.18970736629667 || bata_T : 0.2341220423412204
------------------------------------------------------------
Iter 12-th result :
error rate : 0.19022415940224158 || bata_T : 0.2349096501345636
------------------------------------------------------------
Iter 13-th result :
```

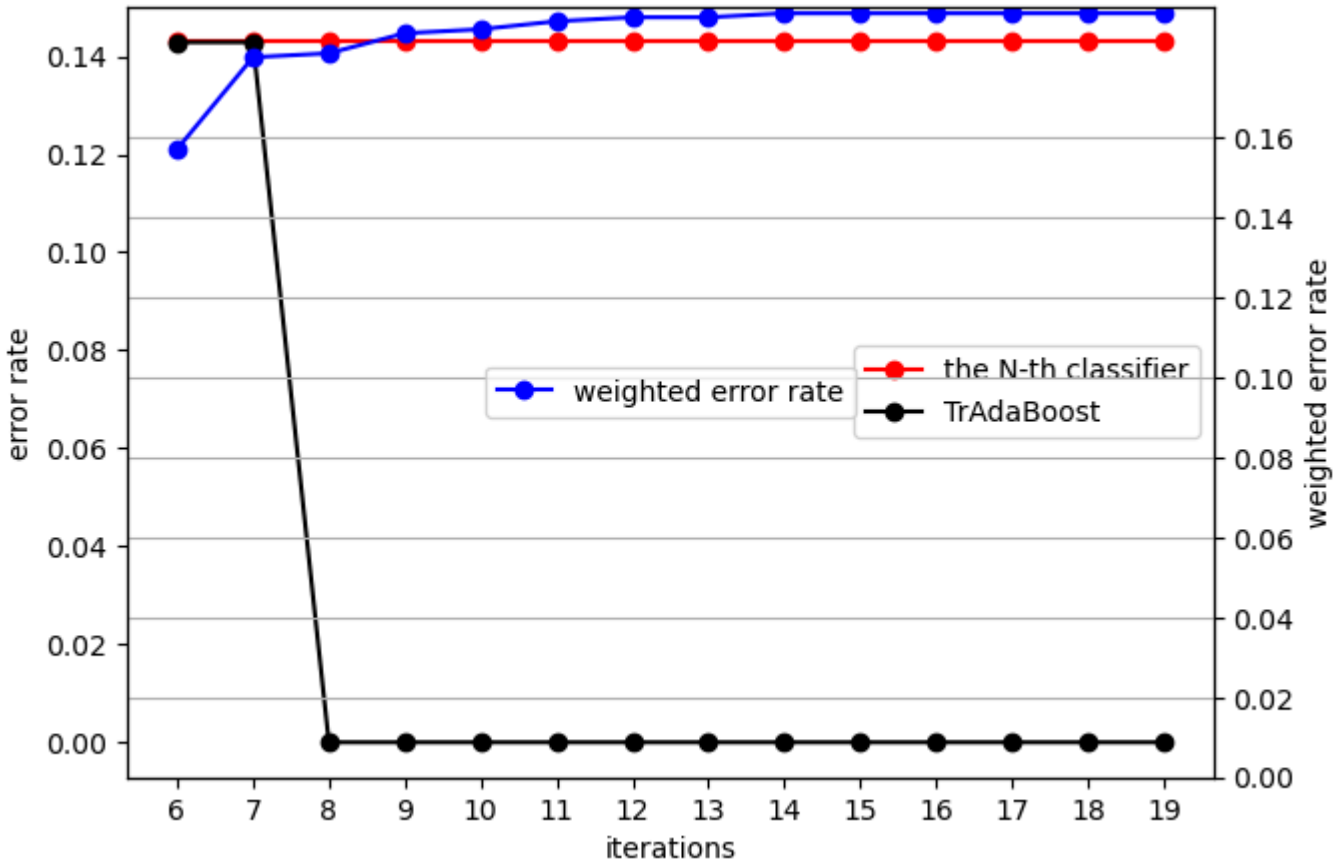
```
error rate : 0.19050365244136874 || bata_T : 0.23533602469722165
---------------------------------------------------------------
Iter 14-th result :
error rate : 0.19069104725718364 || bata_T : 0.23562206572769964
---------------------------------------------------------------
Iter 15-th result :
error rate : 0.19080105633802824 || bata_T : 0.23579004623334252
---------------------------------------------------------------
Iter 16-th result :
error rate : 0.1908711812165715 || bata_T : 0.23589714861912503
---------------------------------------------------------------
Iter 17-th result :
error rate : 0.19091367430396053 || bata_T : 0.23596205774423593
---------------------------------------------------------------
Iter 18-th result :
error rate : 0.19094024787094088 || bata_T : 0.23600265291713904
---------------------------------------------------------------
Iter 19-th result :
error rate : 0.1909565477846002 || bata_T : 0.2360275548386243
---------------------------------------------------------------
TrAdaBoost is done
===============================================================
The prediction labels of test data are :
[0. 0. 0. 1. 1.]
```

In [251…

```
fig = plt.figure(figsize=(7,5))
ax1 = fig.add_subplot(111)
ax1.plot(range(6,20),misclassify_list[5:19],'o-',color="red",label ='the N-th classifier')
ax1.plot(range(6,20),pre_err[5:19],'o-',color="k",label ='TrAdaBoost')
ax1.set_ylabel('error rate')
ax1.set_xlabel('iterations')


ax2 = ax1.twinx()
ax2.plot(range(6,20),error[5:19],'o-',color="b",label='weighted error rate')
ax2.set_ylabel('weighted error rate')
ax2.set_xlabel('Same')


plt.xticks(range(6,20))
plt.yticks(np.linspace(0,0.16,9))
plt.grid()
ax1.legend(loc=5)
ax2.legend(loc=10)
plt.savefig('iteration number.png',bbox_inches = 'tight',dpi=600)
plt.show()
```



Summary of TrAdaBoost

Advantages

- **Adaptability**: It effectively uses source domain data to enhance the performance of the target domain model, especially when target domain data is limited.
- **Flexibility**: It can handle source and target domains with different data distributions.
- **Robustness**: By adjusting sample weights, it reduces the impact of irrelevant source data on the target domain model.

Disadvantages

- **Risk of negative transfer**: If the source and target domains are not strongly related, it may lead to negative transfer, where source data negatively impacts the target domain model.
- **Computational complexity**: It requires processing both source and target domain data, increasing computational complexity.
- **Parameter selection**: Careful tuning of parameters such as the number of iterations and initial weights is necessary to ensure good model performance.

Introduction of MultiSource-TrAdaBoost

MultiSource-TrAdaBoost is an extension of the TrAdaBoost algorithm designed to handle transfer learning scenarios where multiple source domains are available. While TrAdaBoost focuses on transferring knowledge from a single source domain to a target domain, MultiSource-TrAdaBoost leverages multiple source domains to improve the learning process in the target domain. This makes it particularly useful when there are several source domains that may collectively provide more comprehensive information than a single source domain.

Function Definition and Explanation

In [255…

```
def MultiSourceTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N):
    """Boosting for MultiSource Transfer Learning.

    Please feel free to open issues in the Github : https://github.com/Bin-Cao/TrAdaboost
    or
    contact Bin Cao (bcao@shu.edu.cn)
    in case of any problems/comments/suggestions in using the code.

    Parameters
    ----------
    trans_S : feature matrix of same-distribution training data

    Multi_trans_A : dict, feature matrix of diff-distribution training data
    e.g.,
    Multi_trans_A = {
    'trans_A_1' :  data_1 ,
    'trans_A_2' : data_2 ,
    ......
    }
    data_1 : feature matrix of diff-distribution training dataset 1
    data_2 : feature matrix of diff-distribution training dataset 2

    label_S : label of same-distribution training data, -1 or 1

    Multi_label_A : dict, label of diff-distribution training data, -1 or 1
    e.g.,
    Multi_label_A = {
    'label_A_1' :  label_1 ,
    'label_A_2' : label_2 ,
    ......
    }
    label_1 : label of diff-distribution training dataset 1, -1 or 1
    label_1 : label of diff-distribution training dataset 2, -1 or 1

    test : feature matrix of test data

    N : int, default=20
    the number of weak estimators

    References
    ----------
    .. [1] Yao, Y., & Doretto, G. (2010, June)
    Boosting for transfer learning with multiple sources. IEEE.
    DOI: 10.1109/CVPR.2010.5539857

    """
    # prepare trans_A
    trans_A = list(Multi_trans_A.values())[0]
    if len(Multi_trans_A) == 1:
        pass
    else:
        for i in range(len(Multi_trans_A)-1):
            p = i + 1
            trans_A = np.concatenate((trans_A, list(Multi_trans_A.values())[p]), axis=0)
    # prepare label_A
    label_A = list(Multi_label_A.values())[0]
    if len(Multi_label_A) == 1:
        pass
    else:
        for i in range(len(Multi_label_A)-1):
            p = i + 1
```

```
        label_A = np.concatenate((label_A, list(Multi_label_A.values())[p]), axis=0)

trans_data = np.concatenate((trans_A, trans_S), axis=0)
trans_label = np.concatenate((label_A, label_S), axis=0)

row_A = trans_A.shape[0]
row_S = trans_S.shape[0]
row_T = test.shape[0]

if N >= row_A:
    print('The maximum of iterations should be smaller than ', row_A)

test_data = np.concatenate((trans_data, test), axis=0)

# Initialize the weights
weights_A = np.ones([row_A, 1]) / row_A
weights_S = np.ones([row_S, 1]) / row_S
# one-dim column in the shape of ((row_A+row_S),1), column vector
weights = np.concatenate((weights_A, weights_S), axis=0)

alpha_S = 0.5 * np.log((1 + np.sqrt(2 * np.log(row_A / N))))

# Save prediction labels and bata_t
alpha_T = np.zeros([1, N])
result_label = np.ones([row_A + row_S + row_T, N])
# output label
predict = np.zeros([row_T])
print ('params initial finished.')
print('='*60)

trans_data = np.asarray(trans_data, order='C')
trans_label = np.asarray(trans_label, order='C')
test_data = np.asarray(test_data, order='C')

error_rate_list = []
for i in range(N):
    weights = calculate_ratio_weight(weights)

    result_label[:, i], error_rate , Source_index, start = Multi_train_classifier(Multi_t

    # Avoiding overfitting
    if error_rate <= 1e-10:
        N = i
        break

    error_rate_list.append(error_rate)
    alpha_T[0, i] = 0.5 * np.log((1 - error_rate) / error_rate)
    print('Iter {}-th result :'.format(i))
    print('The {}-th diff-distribution training dataset is chosen to transfer'.format(Sou
    print('error rate :', error_rate, '|| alpha_T :', 0.5 * np.log((1 - error_rate) / err
    print('-'*60)

    # Changing the data weights of same-distribution training data
    for j in range(row_S):
        weights[row_A + j] = weights[row_A + j] * np.exp(alpha_T[0, i] * np.abs(result_la
    # Changing the data weights of diff-distribution training data
    for j in range( len(list(Multi_trans_A.values())[Source_index]) ):
        loc = start + j
        weights[loc] = weights[loc] * np.exp(-alpha_S * np.abs(result_label[loc, i] - lab

for i in range(row_T):
    res_ = np.sum(result_label[row_A + row_S + i, :] * alpha_T[0, :])
    if res_ >= 0:
        predict[i] = 1
    else:
        predict[i] = -1
```

```
    print("MultiSourceTrAdaBoost is done")
    print('='*60)
    print('The prediction labels of test data are :')
    print(predict)
    return predict, np.round(np.array(error_rate_list),3)


def calculate_ratio_weight(weights):
    total = np.sum(weights)
    return np.asarray(weights / total, order='C')


def train_classifier(trans_data, trans_label, test_data, ratio_weight):
    clf = tree.DecisionTreeClassifier(criterion="gini", max_depth = 2, max_features="log2", s
    clf.fit(trans_data, trans_label, sample_weight=ratio_weight[:, 0])
    return clf.predict(test_data)

def Multi_train_classifier(Multi_trans_A,label_S, trans_data, trans_label, test_data, weights
    _result_label = np.ones([len(test_data), len(Multi_trans_A)])
    error_record = []
    start_record = []
    start = 0
    for item in range(len(Multi_trans_A)):
        start_record.append(start)
        sub_dataset = list(Multi_trans_A.values())[item]
        data_dim = len(sub_dataset)
        # train a classifier with the 'item'-th data source
        _trans_data = np.concatenate((trans_data[start : start + data_dim], trans_data[row_A:
        _trans_label = np.concatenate((trans_label[start : start + data_dim], trans_label[row
        _ratio_weight = np.concatenate((weights[start : start + data_dim], weights[row_A:row_
        _result_label[:, item] = train_classifier(_trans_data, _trans_label, test_data, _ratio
        start += data_dim
        # cal error rate
        _error_rate = calculate_error_rate(label_S, _result_label[row_A:row_A + row_S, item],
        if _error_rate > 0.5:
            _error_rate = 1 - _error_rate
            # for a binary classifier
            # reverse the prediction label -1 to 1; 1 to -1.
            pre_labels = copy.deepcopy(_result_label[:, item])
            _result_label[:, item] = -pre_labels
        error_record.append(_error_rate)
    error_record = np.array(error_record)
    # choise the best classifier
    classifier_index = np.random.choice(np.flatnonzero(error_record == error_record.min()))
    return _result_label[:,classifier_index], error_record[classifier_index], classifier_inde

def calculate_error_rate(label_R, label_P, weight):
    total = np.sum(weight)
    return np.sum(weight[:, 0] / total * sign(label_R, label_P))

def sign(label_R, label_P):
    _res = label_R - label_P
    for j in range(len(label_R)):
        if _res[j] != 0:
            _res[j]=1
    return _res
```

1. Parameters

- `trans_S` : Feature matrix of the target domain training data.
- `Multi_trans_A` : Dictionary of feature matrices of multiple source domain training data.
- `label_S` : Labels of the target domain training data.
- `Multi_label_A` : Dictionary of labels of multiple source domain training data.

- `test` : Feature matrix of the test data.
- `N` : Number of weak estimators (iterations).

2. Prepare Data

Combine the feature matrices and labels from all source domains into a single array.

```
trans_A = list(Multi_trans_A.values())[0]
if len(Multi_trans_A) > 1:
    for i in range(len(Multi_trans_A) - 1):
        p = i + 1
        trans_A = np.concatenate((trans_A, list(Multi_trans_A.values())[p]), axis=0)
# Prepare label_A
label_A = list(Multi_label_A.values())[0]
if len(Multi_label_A) > 1:
    for i in range(len(Multi_label_A) - 1):
        p = i + 1
        label_A = np.concatenate((label_A, list(Multi_label_A.values())[p]), axis=0)
# Combine source and target domain data
trans_data = np.concatenate((trans_A, trans_S), axis=0)
trans_label = np.concatenate((label_A, label_S), axis=0)
```

3. Initialize Weights

Assign equal weights to all data points in the source and target domains.

```
row_A = trans_A.shape[0]
row_S = trans_S.shape[0]
row_T = test.shape[0]
# Initialize weights
weights_A = np.ones([row_A, 1]) / row_A
weights_S = np.ones([row_S, 1]) / row_S
weights = np.concatenate((weights_A, weights_S), axis=0)
```

4. Iterative Training

Train weak classifiers iteratively, updating weights based on classifier performance.

```
alpha_S = 0.5 * np.log((1 + np.sqrt(2 * np.log(row_A / N))))
alpha_T = np.zeros([1, N])
result_label = np.ones([row_A + row_S + row_T, N])
predict = np.zeros([row_T])
for i in range(N):
    weights = calculate_ratio_weight(weights)
    result_label[:, i], error_rate, Source_index, start = Multi_train_classifier(
        Multi_trans_A, label_S, trans_data, trans_label, test_data, weights, row_A,
row_S
    )
    if error_rate <= 1e-10:
        N = i
        break
    alpha_T[0, i] = 0.5 * np.log((1 - error_rate) / error_rate)
    for j in range(row_S):
        weights[row_A + j] = weights[row_A + j] * np.exp(alpha_T[0, i] *
np.abs(result_label[row_A + j, i] - label_S[j]))
    for j in range(len(list(Multi_trans_A.values())[Source_index])):
        loc = start + j
        weights[loc] = weights[loc] * np.exp(-alpha_S * np.abs(result_label[loc, i]
- label_A[loc]))
```

5. Combine Predictions

Combine predictions from all classifiers to make final predictions on the test data.

```
for i in range(row_T):
    res_ = np.sum(result_label[row_A + row_S + i, :] * alpha_T[0, :])
    if res_ >= 0:
        predict[i] = 1
    else:
        predict[i] = -1
return predict
```

6. Define Multi_train_classifier and other helper functions

Experiment

1. Load Data

In [259…

```
# same-distribution training data
train_data = pd.read_csv('M_Sdata.csv')
# two diff-distribution training data
A1_train_data = pd.read_csv('M_Adata1.csv')
A2_train_data = pd.read_csv('M_Adata2.csv')
# test data
test_data = pd.read_csv('M_Tdata.csv')

Multi_trans_A = {
'trans_A_1' : A1_train_data.iloc[:,:-1],
'trans_A_2' : A2_train_data.iloc[:,:-1]
}
Multi_label_A = {
'label_A_1' :  A1_train_data.iloc[:,-1] ,
'label_A_2' :  A2_train_data.iloc[:,-1] ,
}
trans_S = train_data.iloc[:,:-1]
label_S = train_data.iloc[:, -1]
test = test_data.iloc[:,:-1]
```

2. Data Inspection

In [261…

```
print("Multi_trans_A: ")
print(Multi_trans_A, "\n")
print("Multi_label_A: ")
print(Multi_label_A, "\n")
print("trans_S: ")
print(trans_S, "\n")
print("label_S: ")
print(label_S, "\n")
print("test: ")
print(test)
```

```
Multi_trans_A:
{'trans_A_1':         Cr       1/T     ln(t)    DO
0   10.380  0.001293  4.605170  8000
1   10.380  0.001293  5.521461  8000
2   10.380  0.001293  6.214608  8000
3   10.380  0.001293  6.907755  8000
4   11.022  0.001083  5.298317     0
5   11.022  0.001083  5.991465     0
6   11.022  0.001083  6.684612     0
7   11.022  0.001083  6.396930     0
8   11.022  0.001083  6.907755     0
9   11.160  0.001293  4.605170  8000
10  11.160  0.001293  5.521461  8000
11  11.160  0.001293  6.214608  8000
12  11.160  0.001293  6.907755  8000
13  11.430  0.001293  4.605170  8000
14  11.430  0.001293  5.521461  8000
15  11.430  0.001293  6.214608  8000
16  11.430  0.001293  6.907755  8000
17  11.810  0.001293  4.605170  8000
18  11.810  0.001293  5.521461  8000
19  11.810  0.001293  6.214608  8000
20  11.810  0.001293  6.907755  8000, 'trans_A_2':          Cr       1/T     ln(t)    DO
0   12.9510  0.001215  5.298317   200
1   12.9510  0.001215  5.991465   200
2   12.9510  0.001215  6.396930   200
3   12.9510  0.001215  6.684612   200
4   12.9510  0.001215  6.907755   200
5   14.1100  0.001293  4.605170  8000
6   14.1100  0.001293  5.521461  8000
7   14.1100  0.001293  6.214608  8000
8   14.1100  0.001293  6.907755  8000
9   16.7300  0.001293  5.298317     0
10  16.7300  0.001293  5.991465     0
11  16.7300  0.001293  6.396930     0
12  16.7300  0.001293  6.684612     0
13  16.7300  0.001293  6.907755     0
14  17.0882  0.001215  4.700480     0
15  17.0882  0.001215  5.298317   100
16  17.0882  0.001215  6.131226     0
17  17.0882  0.001215  6.396930     0
18  17.0882  0.001215  5.298317   300
19  17.0882  0.001215  5.991465   100
20  17.0882  0.001215  6.659294     0
21  17.0882  0.001215  5.991465   300
22  17.0882  0.001215  6.907755     0
23  17.0882  0.001215  6.396930   100
24  17.0882  0.001215  5.298317  2000
25  17.0882  0.001215  6.684612   100
26  17.0882  0.001215  6.907755   100
27  17.0882  0.001215  6.396930   300
28  17.0882  0.001215  5.991465  2000}

Multi_label_A:
{'label_A_1': 0    -1
1     1
2     1
3     1
4     1
5     1
6     1
7     1
8     1
9    -1
10    1
11    1
```

```
12    1
13   -1
14    1
15    1
16    1
17   -1
18    1
19    1
20    1
Name: weightgain, dtype: int64, 'label_A_2': 0     1
1     1
2     1
3     1
4     1
5    -1
6     1
7     1
8     1
9    -1
10   -1
11   -1
12   -1
13   -1
14   -1
15   -1
16   -1
17   -1
18   -1
19   -1
20   -1
21   -1
22   -1
23    1
24    1
25    1
26    1
27    1
28    1
Name: weightgain, dtype: int64}

trans_S:
         Cr       1/T     ln(t)  DO
0   17.2690  0.001293  5.298317   8
1   17.2690  0.001293  5.598422   8
2   17.2690  0.001293  6.214608   8
3   17.2690  0.001293  5.940171   8
4   17.2690  0.001215  6.214608   8
5   17.2690  0.001215  4.605170   8
6   18.0596  0.001293  6.907755  10
7   18.0596  0.001293  7.170120  10
8   18.0596  0.001293  7.600902  10
9   18.0596  0.001145  5.768321  25
10  18.0596  0.001145  6.445720  25
11  18.0596  0.001145  6.907755  25
12  18.2693  0.001486  5.541264   8
13  18.2693  0.001486  3.688879   8
14  18.2693  0.001486  6.214608   8
15  18.2693  0.001486  6.363028   8
16  18.2693  0.001293  6.214608   8

label_S:
0    -1
1    -1
2    -1
3    -1
4     1
```

```
5     1
6    -1
7    -1
8     1
9     1
10    1
11    1
12   -1
13   -1
14   -1
15   -1
16   -1
Name: weightgain, dtype: int64

test:
        Cr       1/T     ln(t)  DO
0  17.2690  0.001215  5.298317   8
1  17.2690  0.001215  5.736572   8
2  17.2690  0.001215  5.991465   8
3  18.0596  0.001293  5.857933  10
4  18.0596  0.001293  3.401197  25
5  18.0596  0.001293  6.522093  10
6  18.0596  0.001293  5.768321  25
7  18.0596  0.001293  6.445720  25
```

3. Prediction

In [263…

```
N = 4
MultiSourceTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N,)
```

```
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[ 1.  1.  1. -1. -1. -1. -1. -1.]
```

Out[263…

```
(array([ 1.,  1.,  1., -1., -1., -1., -1., -1.]),
 array([0.059, 0.125, 0.25 , 0.19 ]))
```

4. Visualization

In [265…

```
N_iter = 21
pre_err = []
for i in range(N_iter):
    N = i+1
    pre,_ = MultiSourceTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, trans_S, N)
    pre_err.append(sum(abs(pre - label_S)/2)/len(trans_S))
```

```
pred, error = MultiSourceTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N_i
```

```
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1.  1.  1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
```

```
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117648 || alpha_T : 0.4028125819933177
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117648 || alpha_T : 0.4028125819933177
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.2201559197191664
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
```

```
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3916666666666667 || alpha_T : 0.22015591971916618
------------------------------------------------------------
Iter 7-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958913 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117648 || alpha_T : 0.4028125819933177
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
```

```
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095890 || alpha_T : 0.1800013670157036
------------------------------------------------------------
Iter 8-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434344 || alpha_T : 0.1320757875207932
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.25000000000000006 || alpha_T : 0.5493061443340548
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3916666666666666 || alpha_T : 0.2201559197191664
------------------------------------------------------------
Iter 7-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095890 || alpha_T : 0.1800013670157036
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.42441860465116277 || alpha_T : 0.15233020449309942
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434345 || alpha_T : 0.1320757875207931
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428575 || alpha_T : 0.11659694358385547
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
```

```
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.1904761904761905 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3916666666666667 || alpha_T : 0.22015591971916618
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44799999999999995 || alpha_T : 0.10437740693105518
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117648 || alpha_T : 0.4028125819933177
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.42441860465116277 || alpha_T : 0.15233020449309942
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434343 || alpha_T : 0.13207578752079352
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.448 || alpha_T : 0.1043774069310551
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3916666666666666 || alpha_T : 0.2201559197191664
------------------------------------------------------------
Iter 7-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958896 || alpha_T : 0.18000136701570377
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.42441860465116266 || alpha_T : 0.15233020449309967
------------------------------------------------------------
Iter 9-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434343 || alpha_T : 0.13207578752079352
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428575 || alpha_T : 0.11659694358385547
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.44800000000000006 || alpha_T : 0.10437740693105492
------------------------------------------------------------
Iter 12-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.45289855072463775 || alpha_T : 0.0944830497563115
------------------------------------------------------------
Iter 13-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.45695364238410596 || alpha_T : 0.08630637133349685
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619047 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44799999999999995 || alpha_T : 0.10437740693105518
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.456953642384106 || alpha_T : 0.08630637133349665
------------------------------------------------------------
Iter 14-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853666 || alpha_T : 0.07943494787945221
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117648 || alpha_T : 0.4028125819933177
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.410958904109589 || alpha_T : 0.1800013670157036
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.42441860465116277 || alpha_T : 0.15233020449309942
------------------------------------------------------------
Iter 9-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434343 || alpha_T : 0.13207578752079352
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.448 || alpha_T : 0.1043774069310551
------------------------------------------------------------
Iter 12-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
Iter 13-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4569536423841061 || alpha_T : 0.08630637133349665
------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853666 || alpha_T : 0.07943494787945221
------------------------------------------------------------
Iter 15-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819215 || alpha_T : 0.07357882216814372
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511629 || alpha_T : 0.1523302044930992
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428564 || alpha_T : 0.11659694358385583
------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44799999999999984 || alpha_T : 0.10437740693105536
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45695364238410596 || alpha_T : 0.08630637133349685
------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4603658536585366 || alpha_T : 0.07943494787945231
------------------------------------------------------------
Iter 15-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4632768361581922 || alpha_T : 0.07357882216814354
------------------------------------------------------------
Iter 16-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4657894736842106 || alpha_T : 0.06852812323397899
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.1904761904761905 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.3916666666666667 || alpha_T : 0.22015591971916618
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958913 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428564 || alpha_T : 0.11659694358385583
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4479999999999999 || alpha_T : 0.10437740693105518
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45289855072463775 || alpha_T : 0.0944830497563115
------------------------------------------------------------
Iter 13-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4569536423841059 || alpha_T : 0.08630637133349703
------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853655 || alpha_T : 0.0794349478794525
------------------------------------------------------------
Iter 15-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819204 || alpha_T : 0.07357882216814392
------------------------------------------------------------
Iter 16-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4657894736842105 || alpha_T : 0.06852812323397918
------------------------------------------------------------
Iter 17-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4679802955665025 || alpha_T : 0.06412716776183942
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
```

```
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.125 || alpha_T : 0.9729550745276566
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619044 || alpha_T : 0.7234594914681628
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3088235294117647 || alpha_T : 0.4028125819933178
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.36170212765957444 || alpha_T : 0.2839920188029697
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.42441860465116277 || alpha_T : 0.15233020449309942
------------------------------------------------------------
Iter 9-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434343 || alpha_T : 0.13207578752079352
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.448 || alpha_T : 0.1043774069310551
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4569536423841059 || alpha_T : 0.08630637133349703
------------------------------------------------------------
Iter 14-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853655 || alpha_T : 0.0794349478794525
------------------------------------------------------------
Iter 15-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819215 || alpha_T : 0.07357882216814372
------------------------------------------------------------
```

```
Iter 16-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4657894736842106 || alpha_T : 0.06852812323397899
------------------------------------------------------------
Iter 17-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4679802955665025 || alpha_T : 0.06412716776183942
------------------------------------------------------------
Iter 18-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4699074074074074 || alpha_T : 0.060258012256226186
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.25000000000000006 || alpha_T : 0.5493061443340548
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.1904761904761905 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
Iter 6-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4109589041095891 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428575 || alpha_T : 0.11659694358385547
------------------------------------------------------------
Iter 11-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.44800000000000006 || alpha_T : 0.10437740693105492
------------------------------------------------------------
```

```
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4528985507246377 || alpha_T : 0.09448304975631158
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4569536423841059 || alpha_T : 0.08630637133349703
------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853655 || alpha_T : 0.0794349478794525
------------------------------------------------------------
Iter 15-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4632768361581921 || alpha_T : 0.07357882216814392
------------------------------------------------------------
Iter 16-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46578947368421053 || alpha_T : 0.06852812323397908
------------------------------------------------------------
Iter 17-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4679802955665025 || alpha_T : 0.06412716776183942
------------------------------------------------------------
Iter 18-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46990740740740744 || alpha_T : 0.06025801225622609
------------------------------------------------------------
Iter 19-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.47161572052401757 || alpha_T : 0.05682965923626054
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619052 || alpha_T : 0.7234594914681626
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
```

```
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958913 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511629 || alpha_T : 0.1523302044930992
------------------------------------------------------------
Iter 9-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.43434343434343436 || alpha_T : 0.13207578752079335
------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142857 || alpha_T : 0.11659694358385565
------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44799999999999995 || alpha_T : 0.10437740693105518
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45289855072463764 || alpha_T : 0.09448304975631168
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4569536423841059 || alpha_T : 0.08630637133349703
------------------------------------------------------------
Iter 14-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46036585365853655 || alpha_T : 0.0794349478794525
------------------------------------------------------------
Iter 15-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819215 || alpha_T : 0.07357882216814372
------------------------------------------------------------
Iter 16-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46578947368421053 || alpha_T : 0.06852812323397908
------------------------------------------------------------
Iter 17-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4679802955665025 || alpha_T : 0.06412716776183942
------------------------------------------------------------
Iter 18-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46990740740740733 || alpha_T : 0.06025801225622628
------------------------------------------------------------
Iter 19-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.47161572052401746 || alpha_T : 0.05682965923626074
------------------------------------------------------------
Iter 20-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.47314049586776863 || alpha_T : 0.053770770802093165
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
```

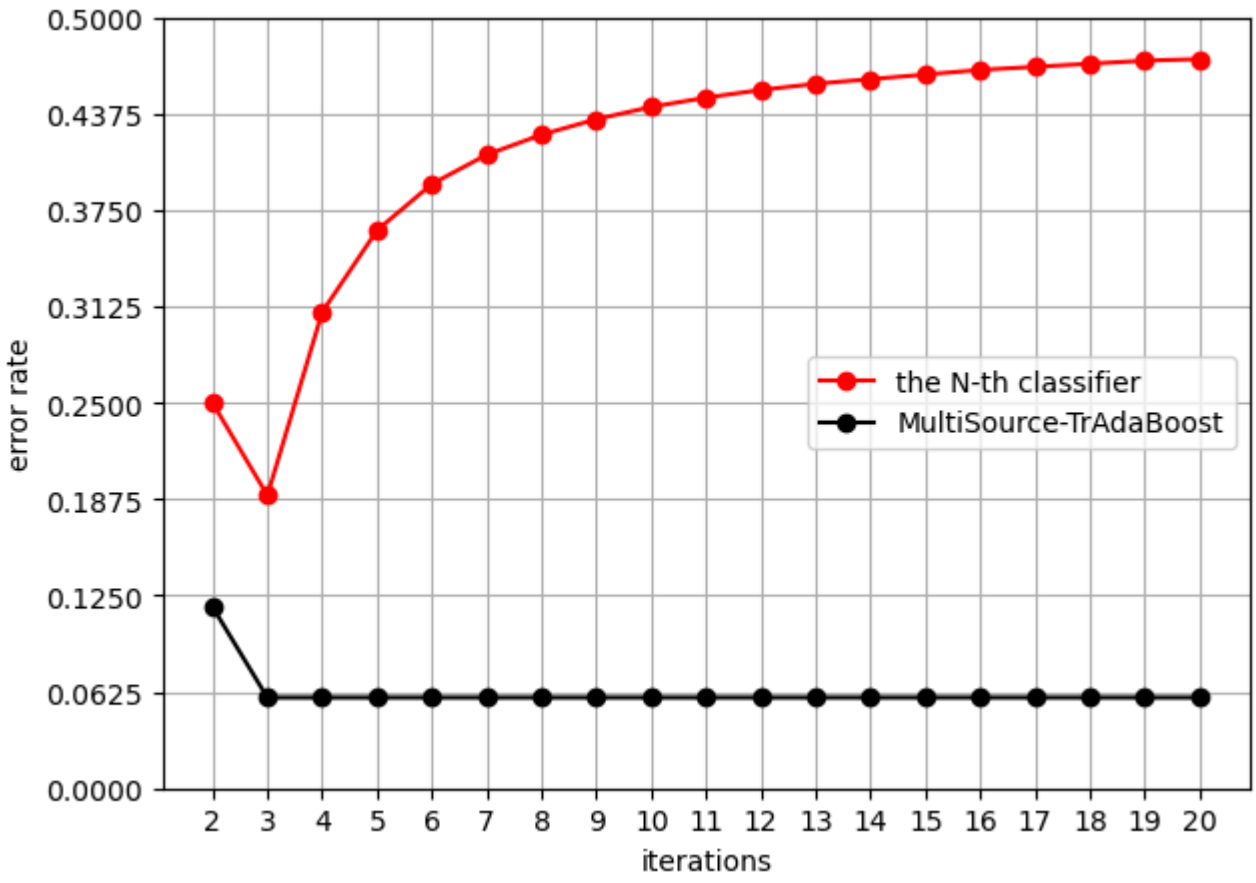
```
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.1904761904761905 || alpha_T : 0.7234594914681627
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595745 || alpha_T : 0.28399201880296965
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958913 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511628 || alpha_T : 0.15233020449309928
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4343434343434343 || alpha_T : 0.13207578752079352
------------------------------------------------------------
Iter 10-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4419642857142856 || alpha_T : 0.11659694358385583
------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44799999999999995 || alpha_T : 0.10437740693105518
------------------------------------------------------------
Iter 12-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45289855072463764 || alpha_T : 0.09448304975631168
------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45695364238410596 || alpha_T : 0.08630637133349685
------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4603658536585366 || alpha_T : 0.07943494787945231
------------------------------------------------------------
Iter 15-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819215 || alpha_T : 0.07357882216814372
------------------------------------------------------------
Iter 16-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.46578947368421053 || alpha_T : 0.06852812323397908
------------------------------------------------------------
Iter 17-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.4679802955665025 || alpha_T : 0.06412716776183942
------------------------------------------------------------
Iter 18-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46990740740740744 || alpha_T : 0.06025801225622609
------------------------------------------------------------
Iter 19-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4716157205240175 || alpha_T : 0.05682965923626064
------------------------------------------------------------
Iter 20-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4731404958677686 || alpha_T : 0.053770770802093366
------------------------------------------------------------
MultiSourceTrAdaBoost is done
============================================================
The prediction labels of test data are :
[ 1.  1.  1. -1. -1. -1. -1. -1.]
```

In [266…

```
fig = plt.figure(figsize=(7,5))
ax1 = fig.add_subplot(111)
ax1.plot(range(2,21),error[2:21],'o-',color="red",label ='the N-th classifier')
ax1.plot(range(2,21),pre_err[2:21],'o-',color="k",label ='MultiSource-TrAdaBoost')
ax1.set_ylabel('error rate')
ax1.set_xlabel('iterations')

plt.xticks(range(2,21))
plt.yticks(np.linspace(0,0.50,9))
plt.grid()
ax1.legend(loc=5)
ax2.legend(loc=10)
plt.savefig('iteration number.png',bbox_inches = 'tight',dpi=600)
plt.show()
```



Key Differences from TrAdaBoost:

1. Multiple Source Domains:

- **TrAdaBoost**: Utilizes knowledge from a single source domain.
- **MultiSource-TrAdaBoost**: Integrates knowledge from multiple source domains, allowing for a richer and more diverse set of information to be transferred.

2. Weight Adjustment Mechanism:

- **TrAdaBoost**: Adjusts weights for instances from a single source domain and the target domain.
- **MultiSource-TrAdaBoost**: Adjusts weights for instances from each source domain individually and combines them with the target domain instances. This allows for more nuanced weight adjustments based on the relevance of each source domain to the target domain.

3. Handling Data Distribution:

- **TrAdaBoost**: Assumes that the source and target domains may have different distributions but focuses on a single source.
- **MultiSource-TrAdaBoost**: Explicitly handles multiple sources with potentially different distributions, making it more flexible in complex transfer learning scenarios.

4. Model Robustness:

TrAdaBoost: Risk of negative transfer if the single source domain is not well-aligned with the target domain. **ultiSource-TrAdaBoost**: Reduces the risk of negative transfer by selecting and combining multiple sources, which can provide a more robust model.

Introduction to Task

TaskTrAdaBoost is a transfer learning algorithm designed to handle multi-source transfer learning tasks. It is an extension of the traditional Adaboost algorithm, specifically tailored to leverage multiple source domains to improve the performance on a target domain. This approach is particularly useful when the target domain has limited labeled data, but there are multiple related source domains with abundant labeled data. Task-TrAdaBoost takes the parameter-transfer approach and both instance-transfer and parameter-transfer approaches belong to the inductive transfer learning. The parameter-transfer approach admits that the target classifier model shares some parameters with the most closely related sources, identifies these shared parameters from various sources and uses them together with the target training data to improve the target classifier learning.

Function Definition and Explanation

In [270…

```
def TaskTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N,gamma):
    """Boosting for MultiSource Transfer Learning.

    Please feel free to open issues in the Github : https://github.com/Bin-Cao/TrAdaboost
    or
    contact Bin Cao (bcao@shu.edu.cn)
    in case of any problems/comments/suggestions in using the code.

    Parameters
    ----------
    trans_S : feature matrix of same-distribution training data

    Multi_trans_A : dict, feature matrix of diff-distribution training data
```

```
e.g.,
Multi_trans_A = {
'trans_A_1' :  data_1 ,
'trans_A_2' : data_2 ,
......
}
data_1 : feature matrix of diff-distribution training dataset 1
data_2 : feature matrix of diff-distribution training dataset 2

label_S : label of same-distribution training data, -1 or 1

Multi_label_A : dict, label of diff-distribution training data, -1 or 1
e.g.,
Multi_label_A = {
'label_A_1' :  label_1 ,
'label_A_2' : label_2 ,
......
}
label_1 : label of diff-distribution training dataset 1, -1 or 1
label_1 : label of diff-distribution training dataset 2, -1 or 1

test : feature matrix of test data

N : int, default=20
the number of weak estimators

gamma : float, for avoiding overfitting

References
----------
.. [1] Yao, Y., & Doretto, G. (2010, June)
Boosting for transfer learning with multiple sources. IEEE.
DOI: 10.1109/CVPR.2010.5539857

"""
weak_classifiers_set = []
for source in range(len(Multi_trans_A)):
    trans_A = list(Multi_trans_A.values())[source]
    label_A = list(Multi_label_A.values())[source]

    trans_A = np.asarray(trans_A, order='C')
    label_A = np.asarray(label_A, order='C')

    # initial weight
    row_A = trans_A.shape[0]
    weights_A = np.ones([row_A, 1]) / row_A

    for j in range(N):
        weights_A = calculate_ratio_weight(weights_A)
        clf = tree.DecisionTreeClassifier(criterion="gini", max_depth = 2,max_features="l
        weak_classifier = clf.fit(trans_A, label_A, sample_weight = weights_A[:, 0])
        pre = weak_classifier.predict(trans_A)
        error_rate = calculate_error_rate(label_A, pre, weights_A)
        alpha = 0.5 * np.log((1-error_rate)/(error_rate+1e-10))
        if error_rate < 0.5 and alpha > gamma:
            weak_classifiers_set.append(weak_classifier)
        else:
            pass
        for j in range(row_A):
            weights_A[j] = weights_A[j] * np.exp(- alpha *  pre[j] * label_A[j])
print('The the set of candidate weak classifiers is initilized and contains {} classifier
print('The phase-I of TaskTrAdaBoost is finished')
print('='*60)

row_S = trans_S.shape[0]
row_T = test.shape[0]
```

```
    test_data = np.concatenate((trans_S, test), axis=0)
    test_data = np.asarray(test_data, order='C')

    # initial weight
    weights_S = np.ones([row_S, 1]) / row_S
    predict = np.zeros([row_T])
    alpha_T = np.zeros([1, N])
    result_label = np.ones([row_S + row_T, N])
    print ('params initial finished.')

    error_rate_list = []
    for k in range(N):
        weights_S = calculate_ratio_weight(weights_S)
        error_rate_set = []
        # save the prediction results of weak classifiers
        _result_label = np.ones([row_S + row_T, len(weak_classifiers_set)])
        for item in range(len(weak_classifiers_set)):
            _result_label[:,item] = weak_classifiers_set[item].predict(test_data)
            _error = calculate_error_rate(label_S, _result_label[0:row_S, item], weights_S)
            if _error > 0.5:
                _error = 1 - _error
                # for a binary classifier
                # reverse the prediction label -1 to 1; 1 to -1.
                pre_labels = copy.deepcopy(_result_label[:, item])
                _result_label[:, item] = - pre_labels
            else:
                pass
            error_rate_set.append(_error)
        error_rate_set = np.array(error_rate_set)

        # choise the best weak classifier and remove it from the set
        classifier_index = np.random.choice(np.flatnonzero(error_rate_set == error_rate_set.m
        result_label[:,k] = _result_label[:,classifier_index]
        error = error_rate_set[classifier_index]
        error_rate_list.append(error)
        if len(weak_classifiers_set) == 0 or error < 1e-10: break
        alpha_T[0, k] = 0.5 * np.log((1 - error) / error)
        weak_classifiers_set.pop(classifier_index)


       # Changing the data weights of same-distribution training data
        for j in range(row_S):
            weights_S[j] = weights_S[j] * np.exp(- alpha_T[0, k] * result_label[j, k] * label

        print('Iter {}-th result :'.format(k))
        print('error rate :', error, '|| alpha_T :', 0.5 * np.log((1 - error) / error) )
        print('-'*60)

    for i in range(row_T):
        res_ = np.sum(result_label[row_S + i, :] * alpha_T[0, :])
        if res_ >= 0:
            predict[i] = 1
        else:
            predict[i] = -1
    print('The phase-II of TaskTrAdaBoost is finished')
    print('='*60)
    print('The prediction labels of test data are :')
    print(predict)
    return predict, np.round(np.array(error_rate_list),3)

def calculate_ratio_weight(weights):
    total = np.sum(weights)
    return np.asarray(weights / total, order='C')

def calculate_error_rate(label_R, label_P, weight):
```

```
    total = np.sum(weight)
    return np.sum(weight[:, 0] / total * sign(label_R, label_P))

def sign(label_R, label_P):
    _res = label_R - label_P
    for j in range(len(label_R)):
        if _res[j] != 0:
            _res[j]=1
    return _res
```

1. Parameters

- `trans_S` : Feature matrix of the target domain (same-distribution training data).
- `Multi_trans_A` : Dictionary containing feature matrices of multiple source domains (different-distribution training data).
- `label_S` : Labels of the target domain.
- `Multi_label_A` : Dictionary containing labels of multiple source domains.
- `test` : Feature matrix of the test data.
- `N` : Number of weak estimators (iterations).
- `gamma` : Threshold to avoid overfitting.

2. Phase I: Initializing Weak Classifiers

In this phase, the algorithm initializes a set of weak classifiers using the source domains.

Iterate over source domains:

- For each source domain, initialize the weights for the samples.
- Train a weak classifier (decision tree) using the weighted samples.
- Calculate the error rate of the classifier.
- If the error rate is below a threshold (0.5) and the alpha value (confidence) is greater than gamma, add the classifier to the set of weak classifiers.
- Update the weights of the samples based on the classifier's performance.

```
for source in range(len(Multi_trans_A)):
    trans_A = list(Multi_trans_A.values())[source]
    label_A = list(Multi_label_A.values())[source]

    trans_A = np.asarray(trans_A, order='C')
    label_A = np.asarray(label_A, order='C')

    # Initial weight
    row_A = trans_A.shape[0]
    weights_A = np.ones([row_A, 1]) / row_A

    for j in range(N):
        weights_A = calculate_ratio_weight(weights_A)
        clf = tree.DecisionTreeClassifier(criterion="gini", max_depth=2,
max_features="log2", splitter="best", random_state=0)
        weak_classifier = clf.fit(trans_A, label_A, sample_weight=weights_A[:, 0])
        pre = weak_classifier.predict(trans_A)
        error_rate = calculate_error_rate(label_A, pre, weights_A)
        alpha = 0.5 * np.log((1 - error_rate) / (error_rate + 1e-10))
        if error_rate < 0.5 and alpha > gamma:
            weak_classifiers_set.append(weak_classifier)
        else:
            pass
```

```
        for j in range(row_A):
            weights_A[j] = weights_A[j] * np.exp(-alpha * pre[j] * label_A[j])
```

3. Phase II: Boosting on Target Domain

In this phase, the algorithm uses the weak classifiers to iteratively improve the performance on the target domain.

Initialize Weights for Target Domain:

- Initialize the weights for the samples in the target domain.
- Combine the target domain data with the test data for prediction.

Iterate Over Weak Classifiers:

- For each iteration, select the best weak classifier from the set.
- Calculate the error rate of the classifier on the target domain.
- Update the weights of the target domain samples based on the classifier's performance.
- Remove the selected classifier from the set.

Predict Test Data:

- Use the selected weak classifiers to predict the labels of the test data.
- Combine the predictions using the alpha values (confidence) to get the final prediction.

Experiment

1. Load Data

In [276…

```
train_data = pd.read_csv('M_Sdata.csv')
# two diff-distribution training data
A1_train_data = pd.read_csv('M_Adata1.csv')
A2_train_data = pd.read_csv('M_Adata2.csv')
# test data
test_data = pd.read_csv('M_Tdata.csv')

Multi_trans_A = {
'trans_A_1' : A1_train_data.iloc[:,:-1],
'trans_A_2' : A2_train_data.iloc[:,:-1]
}
Multi_label_A = {
'label_A_1' :  A1_train_data.iloc[:,-1] ,
'label_A_2' :  A2_train_data.iloc[:,-1] ,
}
trans_S = train_data.iloc[:,:-1]
label_S = train_data.iloc[:, -1]
test = test_data.iloc[:,:-1]
```

2. Prediction and Visualization

In [278…

```
N = 5
gamma = 0.1
pred, error = TaskTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N, gamma,)

N_iter = 21
pre_err = []
for i in range(N_iter):
```

```
    N = i+1
    pre,_ = TaskTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, trans_S, N, gamma,
    pre_err.append(sum(abs(pre - label_S)/2)/len(trans_S))

pred, error = MultiSourceTrAdaBoost(trans_S, Multi_trans_A, label_S, Multi_label_A, test, N_i
```

```
The the set of candidate weak classifiers is initilized and contains 10 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[ 1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 2 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.3529411764705882 || alpha_T : 0.30306790178515786
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 4 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.3529411764705882 || alpha_T : 0.30306790178515786
------------------------------------------------------------
Iter 1-th result :
error rate : 0.46212121212121215 || alpha_T : 0.07590300643400197
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 6 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.3529411764705882 || alpha_T : 0.30306790178515786
------------------------------------------------------------
Iter 1-th result :
error rate : 0.46212121212121215 || alpha_T : 0.07590300643400197
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47171553913645803 || alpha_T : 0.05662937884103338
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 8 classifier
```

```
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.3529411764705882 || alpha_T : 0.30306790178515786
------------------------------------------------------------
Iter 1-th result :
error rate : 0.46212121212121215 || alpha_T : 0.07590300643400197
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47171553913645803 || alpha_T : 0.05662937884103338
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4788119140404382 || alpha_T : 0.04240156481239518
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 10 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 12 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
```

```
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 14 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 16 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 18 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
```

```
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 19 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
```

```
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 20 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 22 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
```

```
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.49212413137148636 || alpha_T : 0.015753040210124657
------------------------------------------------------------
Iter 10-th result :
error rate : 0.4931695486483068 || alpha_T : 0.013661752598970069
------------------------------------------------------------
Iter 11-th result :
error rate : 0.4940750949165553 || alpha_T : 0.011850364855841634
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 23 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
Iter 11-th result :
error rate : 0.49407509491655544 || alpha_T : 0.011850364855841417
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972415 || alpha_T : 0.010280628716336638
------------------------------------------------------------
```

```
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 25 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.49212413137148636 || alpha_T : 0.015753040210124657
------------------------------------------------------------
Iter 10-th result :
error rate : 0.4931695486483068 || alpha_T : 0.013661752598970069
------------------------------------------------------------
Iter 11-th result :
error rate : 0.4940750949165553 || alpha_T : 0.011850364855841634
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972426 || alpha_T : 0.01028062871633642
------------------------------------------------------------
Iter 13-th result :
error rate : 0.4955402217875542 || alpha_T : 0.008919792978321028
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 27 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
```

```
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
Iter 11-th result :
error rate : 0.49407509491655544 || alpha_T : 0.011850364855841417
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972415 || alpha_T : 0.010280628716336638
------------------------------------------------------------
Iter 13-th result :
error rate : 0.49554022178755397 || alpha_T : 0.008919792978321463
------------------------------------------------------------
Iter 14-th result :
error rate : 0.4961302162468544 || alpha_T : 0.007739722047545406
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 28 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
```

```
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.49212413137148636 || alpha_T : 0.015753040210124657
------------------------------------------------------------
Iter 10-th result :
error rate : 0.4931695486483068 || alpha_T : 0.013661752598970069
------------------------------------------------------------
Iter 11-th result :
error rate : 0.4940750949165553 || alpha_T : 0.011850364855841634
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972426 || alpha_T : 0.01028062871633642
------------------------------------------------------------
Iter 13-th result :
error rate : 0.4955402217875542 || alpha_T : 0.008919792978321028
------------------------------------------------------------
Iter 14-th result :
error rate : 0.4961302162468546 || alpha_T : 0.00773972204754486
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358356 || alpha_T : 0.006716185353661329
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 29 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
```

```
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
Iter 11-th result :
error rate : 0.49407509491655544 || alpha_T : 0.011850364855841417
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972415 || alpha_T : 0.010280628716336638
------------------------------------------------------------
Iter 13-th result :
error rate : 0.49554022178755397 || alpha_T : 0.008919792978321463
------------------------------------------------------------
Iter 14-th result :
error rate : 0.4961302162468544 || alpha_T : 0.007739722047545406
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358333 || alpha_T : 0.006716185353661767
------------------------------------------------------------
Iter 16-th result :
error rate : 0.49708589505225254 || alpha_T : 0.005828275887777219
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 30 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.49212413137148636 || alpha_T : 0.015753040210124657
------------------------------------------------------------
Iter 10-th result :
error rate : 0.4931695486483068 || alpha_T : 0.013661752598970069
------------------------------------------------------------
Iter 11-th result :
error rate : 0.4940750949165553 || alpha_T : 0.011850364855841634
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972426 || alpha_T : 0.01028062871633642
```

```
------------------------------------------------------------
Iter 13-th result :
error rate : 0.4955402217875542 || alpha_T : 0.008919792978321028
------------------------------------------------------------
Iter 14-th result :
error rate : 0.4961302162468546 || alpha_T : 0.00773972204754486
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358356 || alpha_T : 0.006716185353661329
------------------------------------------------------------
Iter 16-th result :
error rate : 0.4970858950522526 || alpha_T : 0.005828275887777219
------------------------------------------------------------
Iter 17-th result :
error rate : 0.49747105734781644 || alpha_T : 0.005057928435646537
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 31 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
Iter 11-th result :
error rate : 0.49407509491655544 || alpha_T : 0.011850364855841417
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972415 || alpha_T : 0.010280628716336638
------------------------------------------------------------
Iter 13-th result :
error rate : 0.49554022178755397 || alpha_T : 0.008919792978321463
------------------------------------------------------------
```

```
Iter 14-th result :
error rate : 0.4961302162468544 || alpha_T : 0.007739722047545406
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358333 || alpha_T : 0.006716185353661767
------------------------------------------------------------
Iter 16-th result :
error rate : 0.49708589505225254 || alpha_T : 0.005828275887777219
------------------------------------------------------------
Iter 17-th result :
error rate : 0.4974710573478165 || alpha_T : 0.005057928435646537
------------------------------------------------------------
Iter 18-th result :
error rate : 0.49780525592962954 || alpha_T : 0.004389516332709675
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 32 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.47835497835497837 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.48385358145022317 || alpha_T : 0.03230406941275097
------------------------------------------------------------
Iter 5-th result :
error rate : 0.48603014886417145 || alpha_T : 0.02794697584037156
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540764 || alpha_T : 0.024196984583073004
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.49212413137148636 || alpha_T : 0.015753040210124657
------------------------------------------------------------
Iter 10-th result :
error rate : 0.4931695486483068 || alpha_T : 0.013661752598970069
------------------------------------------------------------
Iter 11-th result :
error rate : 0.4940750949165553 || alpha_T : 0.011850364855841634
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972426 || alpha_T : 0.01028062871633642
------------------------------------------------------------
Iter 13-th result :
error rate : 0.4955402217875542 || alpha_T : 0.008919792978321028
------------------------------------------------------------
Iter 14-th result :
```

```
error rate : 0.4961302162468546 || alpha_T : 0.00773972204754486
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358356 || alpha_T : 0.006716185353661329
------------------------------------------------------------
Iter 16-th result :
error rate : 0.4970858950522526 || alpha_T : 0.005828275887777219
------------------------------------------------------------
Iter 17-th result :
error rate : 0.49747105734781644 || alpha_T : 0.005057928435646537
------------------------------------------------------------
Iter 18-th result :
error rate : 0.4978052559296297 || alpha_T : 0.004389516332709345
------------------------------------------------------------
Iter 19-th result :
error rate : 0.49809525362616797 || alpha_T : 0.0038095111759091603
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1.]
The the set of candidate weak classifiers is initilized and contains 33 classifier
The phase-I of TaskTrAdaBoost is finished
============================================================
params initial finished.
Iter 0-th result :
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
error rate : 0.34375 || alpha_T : 0.32331358246252623
------------------------------------------------------------
Iter 2-th result :
error rate : 0.4783549783549783 || alpha_T : 0.043317115986451146
------------------------------------------------------------
Iter 3-th result :
error rate : 0.4813184130977638 || alpha_T : 0.03738057479630225
------------------------------------------------------------
Iter 4-th result :
error rate : 0.4838535814502233 || alpha_T : 0.03230406941275076
------------------------------------------------------------
Iter 5-th result :
error rate : 0.4860301488641715 || alpha_T : 0.027946975840371455
------------------------------------------------------------
Iter 6-th result :
error rate : 0.4879038683540763 || alpha_T : 0.024196984583073216
------------------------------------------------------------
Iter 7-th result :
error rate : 0.48952013994447197 || alpha_T : 0.020962790190945574
------------------------------------------------------------
Iter 8-th result :
error rate : 0.49091645664127803 || alpha_T : 0.01816908575349449
------------------------------------------------------------
Iter 9-th result :
error rate : 0.4921241313714865 || alpha_T : 0.01575304021012444
------------------------------------------------------------
Iter 10-th result :
error rate : 0.493169548648307 || alpha_T : 0.013661752598969637
------------------------------------------------------------
Iter 11-th result :
error rate : 0.49407509491655544 || alpha_T : 0.011850364855841417
------------------------------------------------------------
Iter 12-th result :
error rate : 0.49485986672972415 || alpha_T : 0.010280628716336638
------------------------------------------------------------
Iter 13-th result :
error rate : 0.49554022178755397 || alpha_T : 0.008919792978321463
```

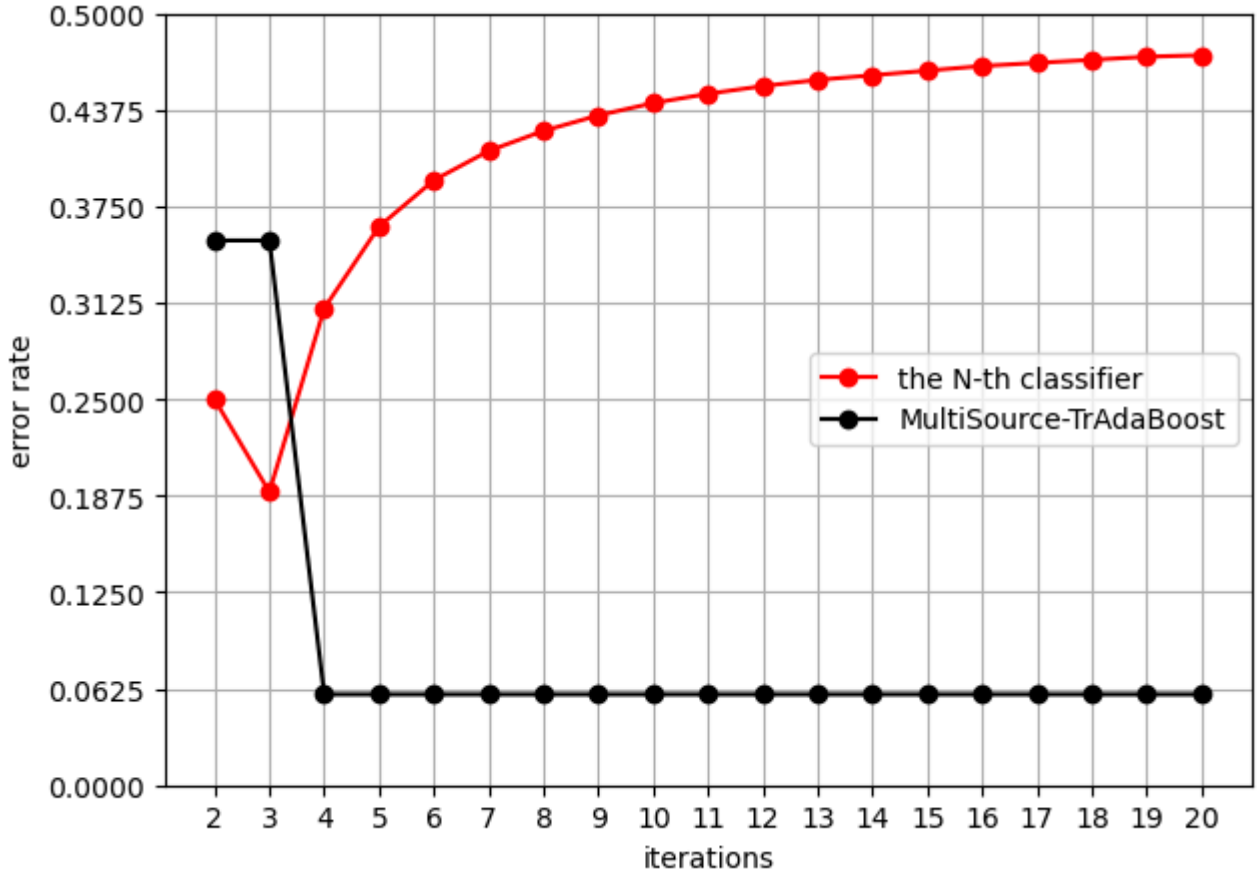
```
------------------------------------------------------------
Iter 14-th result :
error rate : 0.4961302162468544 || alpha_T : 0.007739722047545406
------------------------------------------------------------
Iter 15-th result :
error rate : 0.49664195781358333 || alpha_T : 0.006716185353661767
------------------------------------------------------------
Iter 16-th result :
error rate : 0.49708589505225254 || alpha_T : 0.005828275887777219
------------------------------------------------------------
Iter 17-th result :
error rate : 0.4974710573478165 || alpha_T : 0.005057928435646537
------------------------------------------------------------
Iter 18-th result :
error rate : 0.49780525592962954 || alpha_T : 0.004389516332709675
------------------------------------------------------------
Iter 19-th result :
error rate : 0.4980952536261679 || alpha_T : 0.003809511175909270 5
------------------------------------------------------------
Iter 20-th result :
error rate : 0.49834690912274815 || alpha_T : 0.0033061938010282258
------------------------------------------------------------
The phase-II of TaskTrAdaBoost is finished
============================================================
The prediction labels of test data are :
[-1. -1. -1. -1.  1.  1. -1. -1. -1.  1.  1.  1. -1. -1. -1. -1. -1. -1.]
params initial finished.
============================================================
Iter 0-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.058823529411764705 || alpha_T : 1.3862943611198906
------------------------------------------------------------
Iter 1-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.12500000000000003 || alpha_T : 0.9729550745276565
------------------------------------------------------------
Iter 2-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.24999999999999997 || alpha_T : 0.5493061443340549
------------------------------------------------------------
Iter 3-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.19047619047619052 || alpha_T : 0.7234594914681626
------------------------------------------------------------
Iter 4-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.30882352941176466 || alpha_T : 0.40281258199331793
------------------------------------------------------------
Iter 5-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.3617021276595744 || alpha_T : 0.2839920188029698
------------------------------------------------------------
Iter 6-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.39166666666666666 || alpha_T : 0.22015591971916632
------------------------------------------------------------
Iter 7-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.41095890410958913 || alpha_T : 0.18000136701570338
------------------------------------------------------------
Iter 8-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.4244186046511629 || alpha_T : 0.1523302044930992
------------------------------------------------------------
Iter 9-th result :
The 0-th diff-distribution training dataset is chosen to transfer
```

```
error rate : 0.4343434343434344 || alpha_T : 0.1320757875207932
--------------------------------------------------------------
Iter 10-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.44196428571428575 || alpha_T : 0.11659694358385547
--------------------------------------------------------------
Iter 11-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.448 || alpha_T : 0.1043774069310551
--------------------------------------------------------------
Iter 12-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.45289855072463764 || alpha_T : 0.09448304975631168
--------------------------------------------------------------
Iter 13-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.45695364238410596 || alpha_T : 0.08630637133349685
--------------------------------------------------------------
Iter 14-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4603658536585366 || alpha_T : 0.07943494787945231
--------------------------------------------------------------
Iter 15-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46327683615819215 || alpha_T : 0.07357882216814372
--------------------------------------------------------------
Iter 16-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.4657894736842105 || alpha_T : 0.06852812323397918
--------------------------------------------------------------
Iter 17-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46798029556650245 || alpha_T : 0.06412716776183942
--------------------------------------------------------------
Iter 18-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.46990740740740744 || alpha_T : 0.06025801225622609
--------------------------------------------------------------
Iter 19-th result :
The 0-th diff-distribution training dataset is chosen to transfer
error rate : 0.47161572052401746 || alpha_T : 0.05682965923626074
--------------------------------------------------------------
Iter 20-th result :
The 1-th diff-distribution training dataset is chosen to transfer
error rate : 0.47314049586776863 || alpha_T : 0.053770770802093165
--------------------------------------------------------------
MultiSourceTrAdaBoost is done
==============================================================
The prediction labels of test data are :
[ 1.  1.  1. -1. -1. -1. -1. -1.]
```

In [279…

```
fig = plt.figure(figsize=(7,5))
ax1 = fig.add_subplot(111)
ax1.plot(range(2,21),error[2:21],'o-',color="red",label ='the N-th classifier')
ax1.plot(range(2,21),pre_err[2:21],'o-',color="k",label ='MultiSource-TrAdaBoost')
ax1.set_ylabel('error rate')
ax1.set_xlabel('iterations')

plt.xticks(range(2,21))
plt.yticks(np.linspace(0,0.50,9))
plt.grid()
ax1.legend(loc=5)
ax2.legend(loc=10)
plt.savefig('iteration number.png',bbox_inches = 'tight',dpi=600)
plt.show()
```



Mechanisms:

TrAdaBoost:

- In each iteration, it trains a weak classifier on the combined source and target data.
- Misclassified target samples have their weights increased, while misclassified source samples have their weights decreased.
- The final classifier is a weighted combination of the weak classifiers.

TaskTrAdaBoost:

- Phase I: Initializes a set of weak classifiers by training on each source domain separately.
- Phase II: Selects the best weak classifiers from the set and iteratively improves the target classifier by re-weighting the target samples based on the performance of the weak classifiers.

In [ ]: