Applied Machine Learning

Homework 4

Anand Kamble
Department of Scientific Computing
Florida State University

Environment Setup

In this homework, we implemented the TISP (Thresholded Iterative Soft-thresholding Projection) variable selection method for classification using the hard-thresholding penalty with $\eta = 0$, as described in the course notes. We applied this method to three datasets: Gisette, Dexter, and Madelon.
The environment setup was as follows:

```
conda create -n "homework4" python=3.11
pip install numpy pandas matplotlib scikit-learn
```

Note: Additional packages like `ipykernel` and `jupyter` were used for development purposes.

The code for this homework is available on GitHub:
`https://github.com/anand-kamble/FSU-assignments/blob/main/Machine%20Learning/HW04/main.py`

Implementation of TISP with Hard-Thresholding Penalty

TISP Algorithm Implementation

The TISP algorithm is implemented as follows:

Listing 1: TISP Algorithm Implementation

```
def TISP(X, y, w0, alpha, lambd, num_iter):
    w = w0.copy()
    history = []
    for t in range(num_iter):
        grad = gradient_squared_hinge_loss(X, y, w)
        w = w - alpha * grad
        w = hard_threshold(w, lambd)
        history.append(w.copy())
    return w, history
```

The hard-thresholding operator is defined as:

Listing 2: Hard-Thresholding Operator

```
def hard_threshold(w, lambd):
    w_thresholded = w.copy()
    w_thresholded[np.abs(w_thresholded) < lambd] = 0
    return w_thresholded
```

Data Loading and Preprocessing

We tested the algorithm on the Gisette, Dexter, and Madelon datasets. The datasets were loaded and preprocessed as follows:

Listing 3: Data Loading and Preprocessing

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Initialize variables
y_train = None
X_train = None
X_test = None
y_test = None

# Select the dataset
DATASET = 'DatasetName'  # Options: 'Gisette', 'Dexter', 'Madelon'

if DATASET == 'Gisette':
    train_data_path = './Gisette/gisette_train.data'
    train_labels_path = './Gisette/gisette_train.labels'
    test_data_path = './Gisette/gisette_valid.data'
    test_labels_path = './Gisette/gisette_valid.labels'

    X_train = np.loadtxt(train_data_path)
    y_train = np.loadtxt(train_labels_path)
    X_test = np.loadtxt(test_data_path)
    y_test = np.loadtxt(test_labels_path)

elif DATASET == 'Dexter':
    train_data_path = './dexter/dexter_train.csv'
    train_labels_path = './dexter/dexter_train.labels'
    test_data_path = './dexter/dexter_valid.csv'
    test_labels_path = './dexter/dexter_valid.labels'

    X_train = np.loadtxt(train_data_path, delimiter=',')
    y_train = np.loadtxt(train_labels_path)
    X_test = np.loadtxt(test_data_path, delimiter=',')
    y_test = np.loadtxt(test_labels_path)

elif DATASET == 'Madelon':
    train_data_path = './MADELON/madelon_train.data'
    train_labels_path = './MADELON/madelon_train.labels'
    test_data_path = './MADELON/madelon_valid.data'
    test_labels_path = './MADELON/madelon_valid.labels'

    X_train = np.loadtxt(train_data_path)
    y_train = np.loadtxt(train_labels_path)
    X_test = np.loadtxt(test_data_path)
    y_test = np.loadtxt(test_labels_path)

# Normalize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

The labels y were ensured to be in -1 and $+1$ format.

A. Results for Gisette Dataset

Parameter Selection

The λ values used for the Gisette dataset to achieve the desired number of selected features are:

- $\lambda = 0.0108$
- $\lambda = 0.008$
- $\lambda = 0.0048$
- $\lambda = 0.0028$
- $\lambda = 0.0022$

Misclassification Errors and Selected Features

The misclassification errors and the number of selected features are presented in Table 1.

Table 1: Gisette Dataset Results

λ	Number of Features	Train Error	Test Error
0.0108	8	0.127667	0.146
0.0080	34	0.110667	0.128
0.0048	102	0.056833	0.061
0.0028	318	0.023000	0.023
0.0022	516	0.017333	0.019

Train Misclassification Error vs Iteration Number

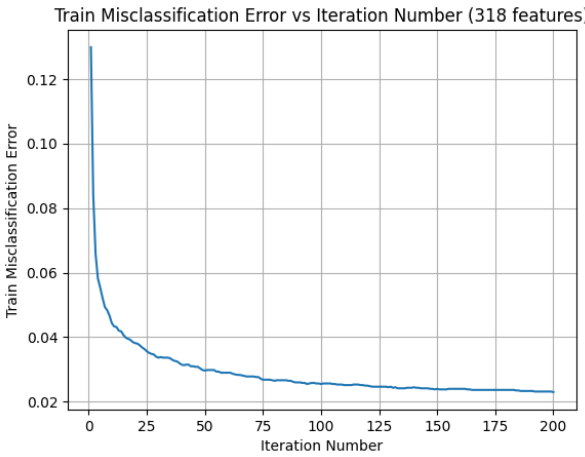

Figure 1: Train Misclassification Error vs Iteration Number (Gisette, 300 features)

Misclassification Error vs Number of Selected Features

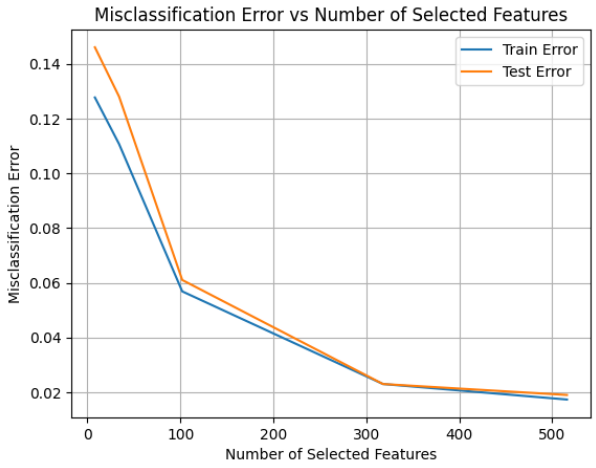

Figure 2: Misclassification Error vs Number of Selected Features (Gisette)

ROC Curves

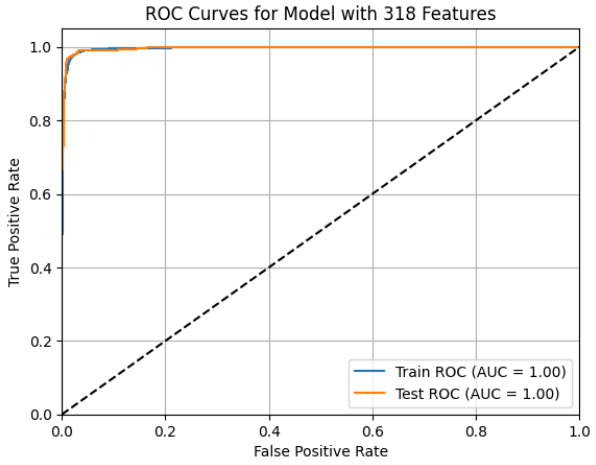

Figure 3: ROC Curves for Model with 300 Features (Gisette)

B. Results for Madelon Dataset

Parameter Selection

The λ values used for the Madelon dataset are:

- $\lambda = 0.0019$
- $\lambda = 0.00095$
- $\lambda = 0.000699$
- $\lambda = 0.000303$
- $\lambda = 0.000001$

Misclassification Errors and Selected Features

The results are presented in Table 2.

Table 2: Madelon Dataset Results

λ	Number of Features	Train Error	Test Error
0.001900	10	0.3860	0.408333
0.000950	30	0.3630	0.431667
0.000699	100	0.3285	0.426667
0.000303	302	0.2735	0.438333
0.000001	500	0.2660	0.421667

Train Misclassification Error vs Iteration Number

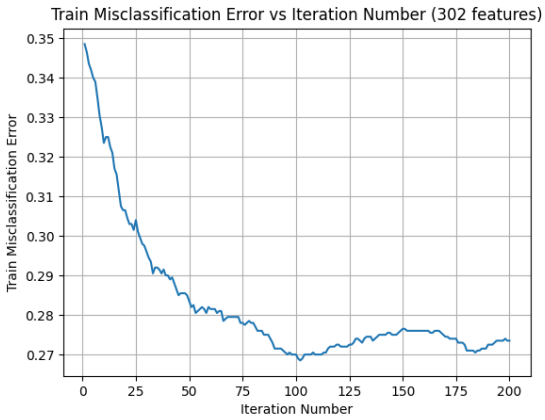

Figure 4: Train Misclassification Error vs Iteration Number (Madelon, 300 features)

Misclassification Error vs Number of Selected Features

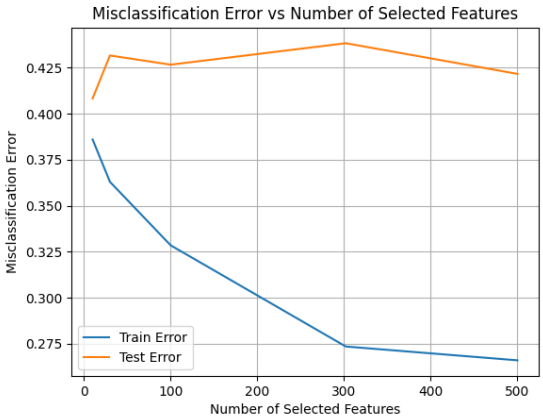

Figure 5: Misclassification Error vs Number of Selected Features (Madelon)

ROC Curves

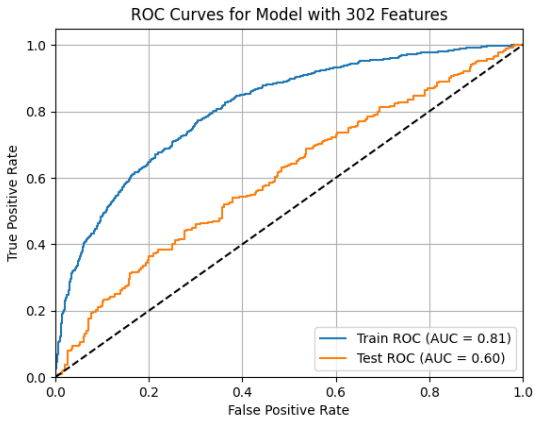

Figure 6: ROC Curves for Model with 300 Features (Madelon)

C. Results for Dexter Dataset

Parameter Selection

The λ values used for the Dexter dataset are:

- $\lambda = 0.006$
- $\lambda = 0.00405$
- $\lambda = 0.0029$
- $\lambda = 0.00216$
- $\lambda = 0.0019$

Misclassification Errors and Selected Features

The results are presented in Table 3.

Table 3: Dexter Dataset Results

λ	Number of Features	Train Error	Test Error
0.00600	10	0.176667	0.183333
0.00405	31	0.066667	0.113333
0.00290	100	0.020000	0.080000
0.00216	301	0.003333	0.070000
0.00190	509	0.000000	0.083333

Train Misclassification Error vs Iteration Number

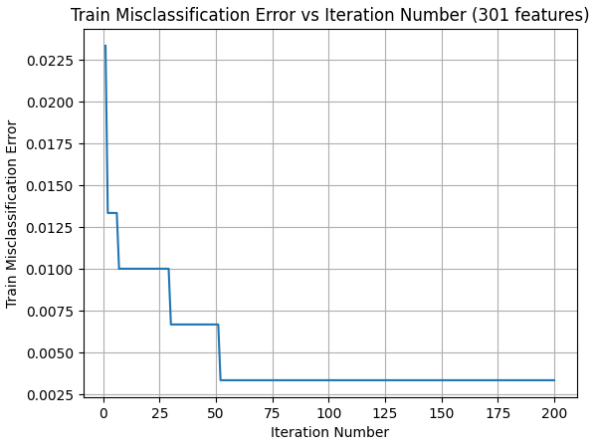

Figure 7: Train Misclassification Error vs Iteration Number (Dexter, 300 features)

Misclassification Error vs Number of Selected Features

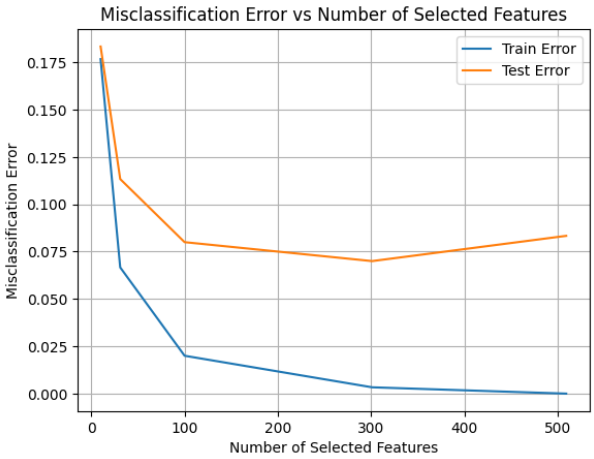

Figure 8: Misclassification Error vs Number of Selected Features (Dexter)

ROC Curves

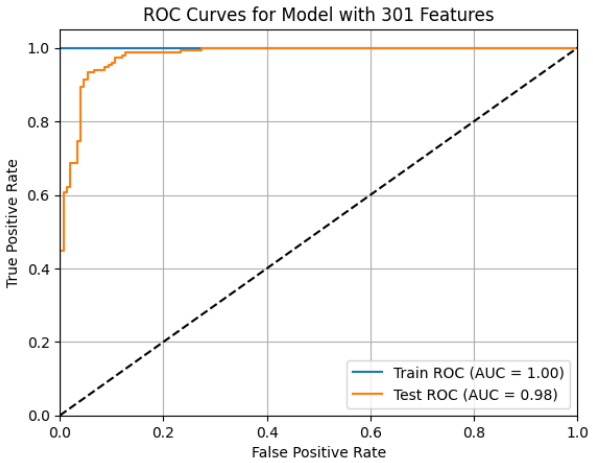

Figure 9: ROC Curves for Model with 300 Features (Dexter)