

2025 Autumn Intro-to-Machine Learning Homework 2

Release Date: 2025/10/14 15:00

Homework 2

- Deadline: **23:59, Oct. 28th (Tue), 2025**
- **Coding (60%)**: Implement linear classification methods by **only** using **numpy**.
 - Submit your code in executable python files (.py).
 - Report the outcome and parameters by screenshots to the questions.
- **Handwritten Questions (40%)**: Answer questions about linear classification.
 - Answer the questions in the report.
 - You **must use the template** and in **digital-typed (no handwritten scan)**
 - In English

Links

- [Questions and Report template](#)
- [Sample code / Dataset](#)

Coding Environment

- Recommded: Python 3.9 or higher
- Tips
 - We recommend you to use **virtual environments** when implementing your homework assignments.
 - Here are some popular virtual environment management tools
 - [uv](#)
 - [Poetry](#)
 - [Conda](#)
 - [Virtualenv](#)

Numpy

- High efficient vector and matrix operations
- Numpy Tutorial: [Link](#)

element-wise
multiply

```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
for i in range(a.shape[0]):
    a[i] *= b[i]
print(a)
# a = [ 4 10 18]
```



```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
a *= b
print(a)
# a = [ 4 10 18]
```

square root

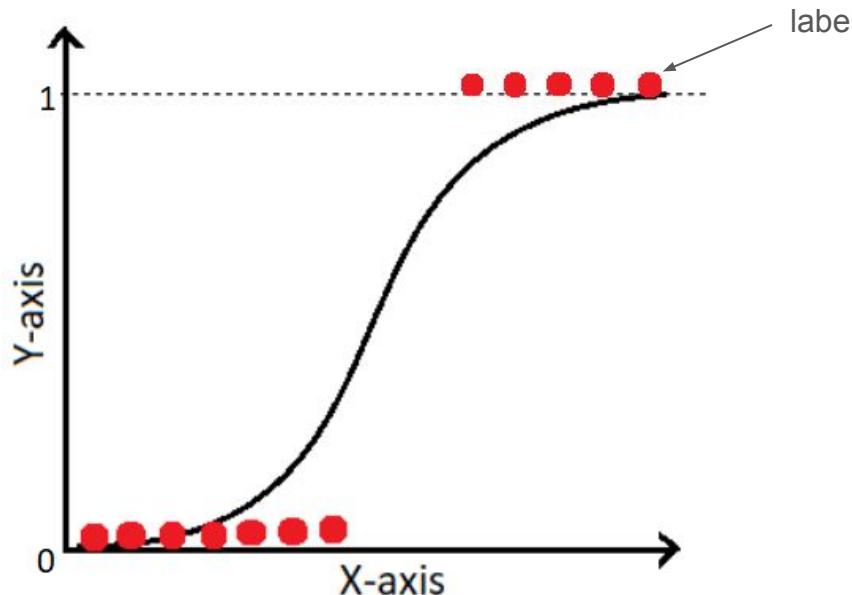
```
import math
a = np.array([1, 4, 9])
for i in range(a.shape[0]):
    a[i] = math.sqrt(a[i])
print(a)
# a = [1 2 3]
```



```
a = np.array([1, 4, 9])
a = np.sqrt(a)
print(a)
# a = [1 2 3]
```

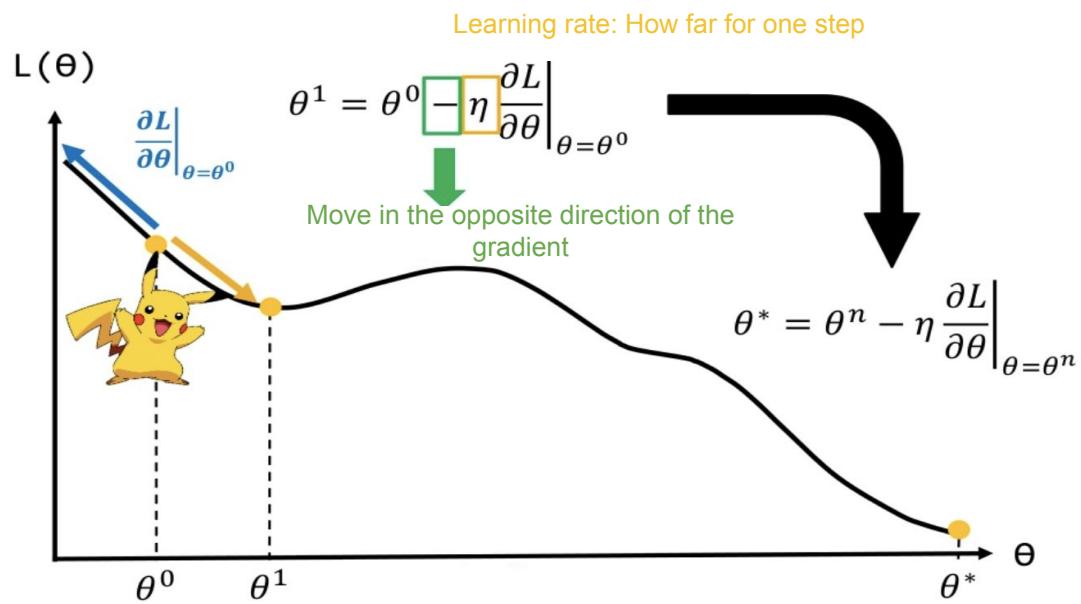
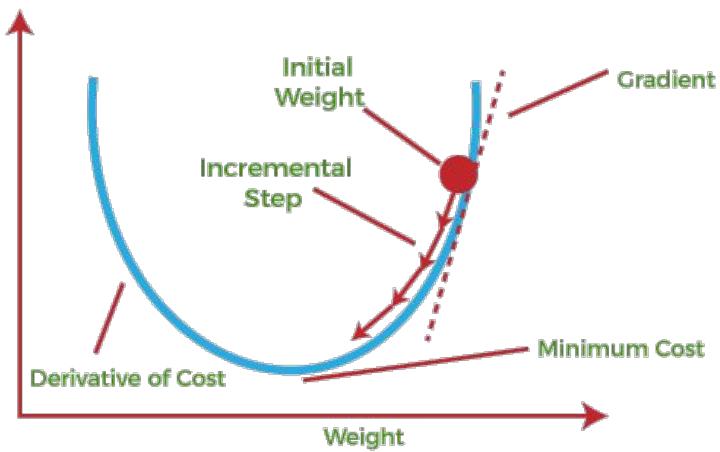
Logistic Regression for Binary classification

- Find the best value of the weights and the intercept of a logistic model



Logistic Regression

- Gradient Descent



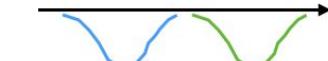
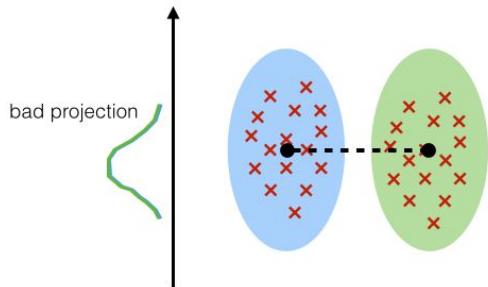
To better understand GD, see <https://www.youtube.com/watch?v=yKKNr-QKz2Q>

Fisher's Linear Discriminant, FLD

- FLD seeks the projection w that gives a large distance between the projected data center (means) while giving a small variance within each class.

LDA:

maximizing the component axes for class-separation



good projection: separates classes well

$$J(\mathbf{W}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

■ Between-class variance

■ Within-class variance

Dataset and Environment

- Sonar dataset - A binary classification problem
- Datapoints
 - 208 (166 for train / 42 for test)
- Features
 - 60 features (already normalized), but only two columns are assigned in this homework.
- Target
 - Rock or Mine (0 / 1)
- Required packages: `numpy`, `pandas`, `matplotlib`, `loguru`, `flake8`, `pytest`, `scikit-learn`

Logistic Regression (25%)

- Requirements
 - Use Gradient Descent to update your model.
 - Use CE (Cross-Entropy) as your loss function.
- Grading Criteria
 - (5%) Show the **hyperparameters** (learning rate and iteration, etc) that you used and the **weights and intercept** of your model.
 - (5%) Show the AUC score of the classification results on the testing set.
 - sklearn is allowed to compute AUC score (only for this)
 - (15%) Show the **accuracy score** of your model **on the testing set**.

Accuracy	Score (pt)
≥ 0.8	15 pt
$\geq 0.75, < 0.8$	10 pt
$\geq 0.7, < 0.75$	5 pt
< 0.7	0 pt

Fisher's Linear Discriminant, FLD (25%)

- Requirements:
 - Implement FLD to project the data from 2-dimensional to 1-dimensional space.
- Criteria:
 - (5%) Show the **mean vectors m_i ($i=0, 1$) of each class**, the **within-class scatter matrix S_w** , and the **between-class scatter matrix S_b** of the training set.
 - (5%) Show the Fisher's linear discriminant w of the training set.

Fisher's Linear Discriminant

- Criteria

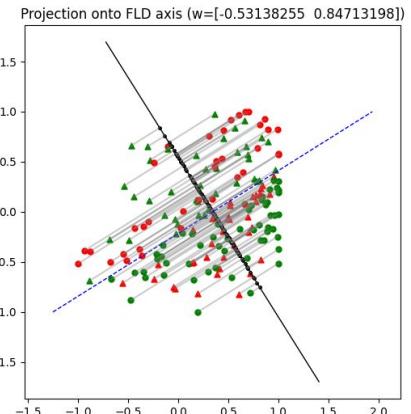
- (15%, Acc=10%, Plot=5%) Obtain predictions for the testing set by measuring the distance between the projected value of the testing data and the projected means of the testing data for the two classes.

- **Plot the projection line** trained on the training set and show the slope and intercept on the title. (gray line)
 - **Plot the Decision boundary** (blue line)
 - **Plot and colorize and mark** data points based on the ground-truth and prediction.
 - **Marker: Circle for y_{true} is 0; Triangle for 1**
 - **Color: Green if the prediction is correct; Red if prediction is incorrect**
 - Project all testing data points onto 1d space.

- Also, **Show the accuracy score** on the testing set.

Accuracy	Score (pt)
≥ 0.72	10 pt
$\geq 0.65, < 0.72$	5 pt
< 0.65	0 pt

Just an example



Code Output

- Do not modify the main function architecture heavily.
- Your code output will look like this

```
2024-10-13 07:28:03.878 | INFO | __main__:fit:44 - Iteration 1, Train-Acc=0.4639
2024-10-13 07:28:04.072 | INFO | __main__:fit:44 - Iteration 1001, Train-Acc=0.6687
2024-10-13 07:28:04.140 | INFO | __main__:fit:44 - Iteration 2001, Train-Acc=0.7108
2024-10-13 07:28:04.242 | INFO | __main__:fit:44 - Iteration 3001, Train-Acc=0.7229
2024-10-13 07:28:04.339 | INFO | __main__:fit:44 - Iteration 4001, Train-Acc=0.7410
2024-10-13 07:28:04.392 | INFO | __main__:main:187 - LR: Weights: [REDACTED], Intercept: [REDACTED]
2024-10-13 07:28:04.393 | INFO | __main__:main:188 - LR: Accuracy=[REDACTED] AUC=[REDACTED]
2024-10-13 07:28:04.395 | INFO | __main__:main:205 - FLD: m0=[REDACTED], m1=[REDACTED]
2024-10-13 07:28:04.396 | INFO | __main__:main:206 - FLD:
Sw=
[REDACTED]

2024-10-13 07:28:04.396 | INFO | __main__:main:207 - FLD:
Sb=
[REDACTED]

2024-10-13 07:28:04.396 | INFO | __main__:main:208 - FLD:
W=
[REDACTED]

2024-10-13 07:28:04.396 | INFO | __main__:main:209 - FLD: Accuracy=[REDACTED]
```

1. [PEP8](#)
2. [Google Python Style](#)

Additional Requirements

Code Check and Verification: Lint the code and show the PyTest results (10%)

- Code linting (**screenshot even if there is no output for fully passed linting**):
 - **-2pt** per warning / error
- Run PyTest: \$ pytest ./test_main.py -s
 - **-5pt** per failed case

```
$ flake8 main.py
./main.py:103:1: W391 blank line at end of file
1     W391 blank line at end of file
```

```
===== test session starts =====
platform linux -- Python 3.9.16, pytest-8.1.1, pluggy-1.4.0
rootdir: /home/seanyu/lectures/nycu/ml-and-pattern-recognition/hw2
configfile: pyproject.toml
collected 2 items

test_main.py (395, 2) (395,)
2024-10-13 07:36:37.581 | INFO    | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2024-10-13 07:36:37.587 | INFO    | test_main:test_fld:45 - accuracy=0.8759
.

===== 2 passed in 3.73s =====
```

Handwritten Questions (40%)

1. (15%)

(*) Using (4.57) and (4.58), derive the result (4.65) for the posterior class probability in the two-class generative model with Gaussian densities, and verify the results (4.66) and (4.67) for the parameters \mathbf{w} and w_0 .

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned} \tag{4.57}$$

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{4.58}$$

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0) \tag{4.65}$$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.66}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \tag{4.67}$$

Handwritten Questions (40%)

2. (10%)

- (a) Give one real-world situation where you would prefer Logistic Regression (LR) over the Perceptron, and explain why.
- (b) Is Logistic Regression actually used for regression (predicting a continuous value)? If not, state what task it really solves and why the name includes “regression.”

3. (15%)

- (a) Why is feature scaling (e.g., standardization or normalization) important in Logistic Regression? Explain **two reasons**.
- (b) If feature scaling is not applied in Logistic Regression, list **three problems** that may occur. Briefly explain.

Report

- Please follow the report template format. (**-5pts** if not use the template)
- [Link](#)

Submission

- Compress your **code** and **report** into a **.zip** file and submit it to E3.
- Report should be written in English. (-5 pts if not English)
- <STUDENT ID>_HW2.zip
 - main.py
 - setup.cfg
 - test_main.py
 - <STUDENT ID>_HW2.pdf (NO .doc, .docx or others format)
- Don't put the data (e.g. train.csv / test.csv) into submission file (-5 pts if you put data into the zipped file)

Other rules

- **Late Policy**: A penalty of **20 points per additional late day.** (**-20pt / delayed.day**)
 - For example, If you get 90 points but delay for two days, your will get only 50 points!
- **No Plagiarism**: You should complete the assignment by yourself. Students engaged in plagiarism will be penalized heavily. Super serious penalty.
 - e.g. -100pt for the assignment or failed this course, etc
 - Report to academic integrity office

AI-Assistant

- Not recommended but no forbidden
- Copy-and-Paste answers from the AI-Assiant will be seen as Plagiarism
 - However, you can have your own answer first then rephrase it by AI-Assiant.
- Some questions might be parts of final exam, make sure you understand the concept



FAQs

- If you have other questions, ask on [**E3 forum**](#) first! We will reply as soon as possible.
 - Also, feel free to write email to TAs (And remember to cc all TAs).

Have Fun!

