Midterm Review

1. **Machine Learning Understanding**

- Supervised Learning vs. unsupervised Learning

Supervised learning has a clear objective such detecting spam email, predicting prices
Unsupervised learning aims to find a pattern, anomalies, or clusters.

- Regression vs. classification

Regression predicts a value, while classification predicts a label.

- Training set vs. test set

The training set is used to train the model, while the test set is used to evaluate the model's performance. We should split the data set into a training set and a test set before the training process. The test set is not used to train the model, so it represents new data.

- How to evaluate a regression model?

MSE (mean-squared-error), MAE (mean-absolute-error), Visualize the model and see how it fits the data.

- How to evaluate a classification model?

Accuracy, Precision, Recall, Confusion Matrix, F-1 score, Visualization of the decision regions..

- Cross validation

Cross validation eliminates the possibility that the good performance only occurs for a specific training set.

- Which model is a proper choice for a regression task: linear regression, polynomial regression
- Which model is a proper choice for a classification task: logistic regression

2. **Machine Learning Models**

- Linear regression
- Polynomial regression
- Logistic regression

For each model, you are expected to explain:

- The mathematical expression for that model
- What cost function can be used to measure the model's performance during the training phase?
- How to find parameter values that correspond to an optimal cost?
- Work on a simple example

Index	(1)	(2)	(3)	(4)
X (input variable)	1	2	3	4

Y (output variable)	4	4.9	6.2	8

Linear Regression:

Intuition: We use a line (a plane, or a hyper-plane) to describe the relationship between the input variables X1, X2, …, Xn and the output variable y

Model expression

$$y = mx + b$$

If there is a single input variable

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

MSE cost function:

$$J(\theta_0, \ldots \theta_n) = \frac{1}{N} \sum (\text{actual output - prediction})^2$$

How to find the minimal cost?

1. Use the normal equation
2. Use the gradient descent method

Polynomial Regression:
Intuition: Use a polynomial curve to describe the relationship between the input and the output.

Model expression: single variable, degree 4

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Model expression: suppose there are two variables x1 and x2, and the largest degree is 3.

$$y = \theta_{00} + \theta_{11} x_1 + \theta_{12} x_2 + \theta_{21} x_1^2 + \theta_{22} x_1 x_2 + \theta_{23} x_2^2 + \theta_{31} x_1^3 + \theta_{32} x_1^2 x_2 + \theta_{33} x_1 x_2^2 + \theta_{34} x_2^3$$

MSE cost function $J(\theta) = \frac{1}{N} \sum$ (actual output - model prediction)^2

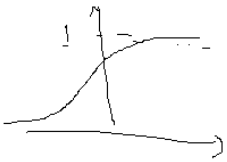
Training algorithm:
1. Using normal equation (with polynomial features)
2. Gradient descent method

Logistic Regression

Intuition: Predict probability distribution over all possible classes. The probability distribution is represented as a linear combination of all inputs, following by a transformation that guarantees the output to be probabilities (logistic function or the softmax function).

Model expression: binary classifier

$$P = \sigma(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n) \qquad \sigma(t) = \frac{1}{1+e^{-t}}$$



Model expression: multiple classes

$t_1, \cdots, t_n$ = linear combinations of input variables

$P_1, \cdots, P_n = \sigma(t_1, \cdots, t_n)$ σ : softmax function

Cross-Entropy Cost function

Training algorithm: gradient descent

3. Python Libraries

- NumPy: NumPy array, mathematical functions, matrix functions
- Pandas: Pandas data frame, data handling, data transformation
- Matplotlib: line plot, scatter plot, histogram
- Sklearn: models, model evaluations

TODO: go over the class notes and summarize the functions we have used.