# Week 5 Discussion

COGS118A Instructional Team

COGS 118A: Supervised Machine Learning Algorithms

## 1 True or False

1. The dot product between two vectors will change in value if one of the vectors gets longer (multiplied by a scalar) but stays pointed in the same direction.

   [True]

   [False]

   **Solution:**

2. Training set error is equal to testing set error plus generalization error; i.e., $e_{\text{training}} = e_{\text{testing}} + e_{\text{generalization}}$

   [True]

   [False]

   **Solution:**

3. For a monotonically increasing function $f(x) \in \mathbb{R}$, $h(x) = \ln f(x)$ is monotonically decreasing.

   [True]

   [False]

   **Solution:**

4. For a monotonically increasing function $f(x) \in \mathbb{R}$, $h(x) = f(x) + e^x$ is monotonically increasing.

   [True]

   [False]

   **Solution:**

# 2 Multiple Choice Questions

1. When evaluating the performance of a classifier, which of the following methods will help you get the best estimate of generalization error when you have relatively few data points

   A. Divide the data into separate training and test sets

   B. Use the same data for training and testing

   C. 5x repeated shuffle-splits

   D. Leave one out cross validation (k-fold cross validation where k=n)

   **Solution:**

2. A function is strictly convex when for all pairs of points $(w_0, w_1)$ on the function, a chord between those points lies ______ the function evaluated between $(w_0, w_1)$. Fill in the blank to make this statement true

   A. above

   B. tangent to

   C. along

   D. below

   **Solution:**

3. Assume an optimization problem of the form $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$. Pick the true statement from the following:

   A. A closed-form solution for $\mathbf{w}^*$ always exists and can be found using the least squares estimation algorithm.

   B. $\mathcal{L}(\mathbf{w})$ is defined as the error of the algorithm on the test set.

   C. Local and global minima will occur at critical points where $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = 0$.

   D. Solving for $\mathbf{w}^*$ requires using gradient descent.

   **Solution:**

4. What is the main reason to use $L_1$ instead of $L_2$ as the error term of a loss function when estimating a regression model?

   A. To apply the gradient descent algorithm in training.

   B. To make the model linear.

   C. To make the model robust against outliers.

   D. To speed up the training process.

   **Solution:**

5. Assume we have a binary classification model:

$$f(\mathbf{x}) = \begin{cases} +1, & \mathbf{w} \cdot \mathbf{x} + b \geq 0, \\ -1, & \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases}$$

   where $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ (feature vector), $b \in \mathbb{R}$ (bias), and $\mathbf{w} = (w_1, w_2) \in \mathbb{R}^2$ (weight vector). The predictions of classifier $f$ and its decision boundary $\mathbf{w} \cdot \mathbf{x} + b = 0$ are shown in Figure 1. Pick the answer below that best matches the drawing in that figure.

A. $\mathbf{w} = (+1, 0), b = -1$.
B. $\mathbf{w} = (-1, 0), b = +1$.
C. $\mathbf{w} = (0, +1), b = +1$.
D. $\mathbf{w} = (0, -1), b = -1$.

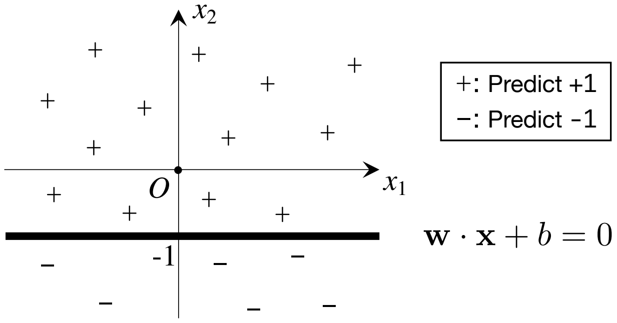
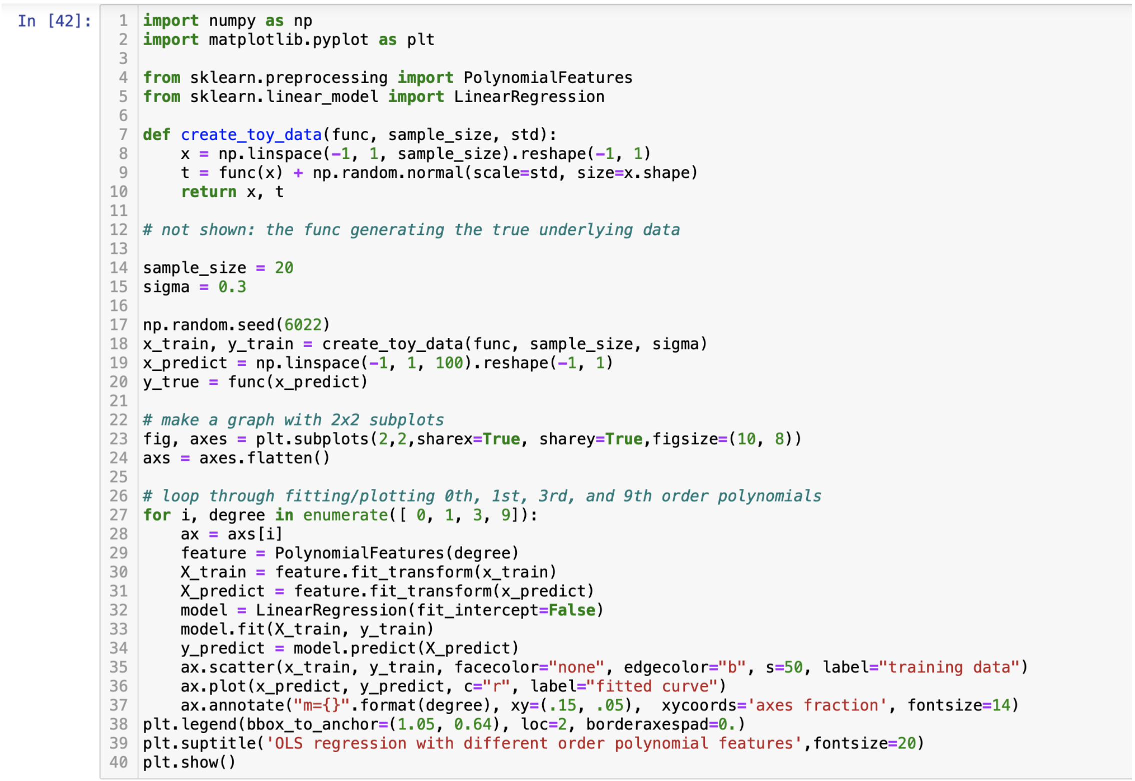


Figure 1: Decision boundary

**Solution:**

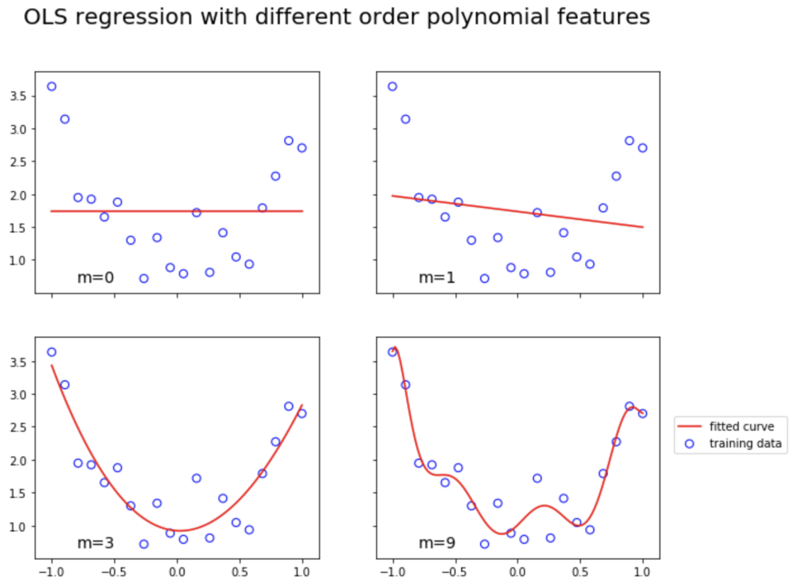
# 3 Polynomial features for OLS

Below is code to perform Ordinary Least Squares regression using polynomial features. The code finds the optimum weight values $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$ using the OLS algorithm. The polynomial features can be built with varying order, represented according to the numeric value of $m$ such that $f(x;\mathbf{w}) = \mathbf{w}_0 + \mathbf{w}_1 x + \mathbf{w}_2 x^2 + \dots \mathbf{w}_m x^m$. The figure below shows 4 different order of polynomials fit to the same training set.



In [42]:

```
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

def create_toy_data(func, sample_size, std):
    x = np.linspace(-1, 1, sample_size).reshape(-1, 1)
    t = func(x) + np.random.normal(scale=std, size=x.shape)
    return x, t

# not shown: the func generating the true underlying data

sample_size = 20
sigma = 0.3

np.random.seed(6022)
x_train, y_train = create_toy_data(func, sample_size, sigma)
x_predict = np.linspace(-1, 1, 100).reshape(-1, 1)
y_true = func(x_predict)

# make a graph with 2x2 subplots
fig, axes = plt.subplots(2,2,sharex=True, sharey=True,figsize=(10, 8))
axs = axes.flatten()

# loop through fitting/plotting 0th, 1st, 3rd, and 9th order polynomials
for i, degree in enumerate([ 0, 1, 3, 9]):
    ax = axs[i]
    feature = PolynomialFeatures(degree)
    X_train = feature.fit_transform(x_train)
    X_predict = feature.fit_transform(x_predict)
    model = LinearRegression(fit_intercept=False)
    model.fit(X_train, y_train)
    y_predict = model.predict(X_predict)
    ax.scatter(x_train, y_train, facecolor="none", edgecolor="b", s=50, label="training data")
    ax.plot(x_predict, y_predict, c="r", label="fitted curve")
    ax.annotate("m={}".format(degree), xy=(.15, .05),  xycoords='axes fraction', fontsize=14)
plt.legend(bbox_to_anchor=(1.05, 0.64), loc=2, borderaxespad=0.)
plt.suptitle('OLS regression with different order polynomial features',fontsize=20)
plt.show()
```



**(a)** Let's say one of the training datapoints is located at $(x = -0.8, y = 2.5)$.

Write the polynomial feature vector for this datapoint for all of the following cases:

$\mathbf{m = 0}$: x= $\left[ \qquad\qquad\qquad\qquad\qquad\qquad \right]$

$\mathbf{m = 1}$: x= $\left[ \qquad\qquad\qquad\qquad\qquad\qquad \right]$

$\mathbf{m = 3}$: x= $\left[ \qquad\qquad\qquad\qquad\qquad\qquad \right]$

**(b)** Given $e_{\text{test}} = \mathcal{L}(\mathbf{w}^*, S_{\text{test}})$ (the loss function using the optimal weights evaluated on test data), which order model do you think is closest to the order of the true generating function?

A. $m = 0$ with $e_{\text{test}} = 130$

B. $m = 1$ with $e_{\text{test}} = 100$

C. $m = 3$ with $e_{\text{test}} = 50$

D. $m = 9$ with $e_{\text{test}} = 80$

**Solution:**

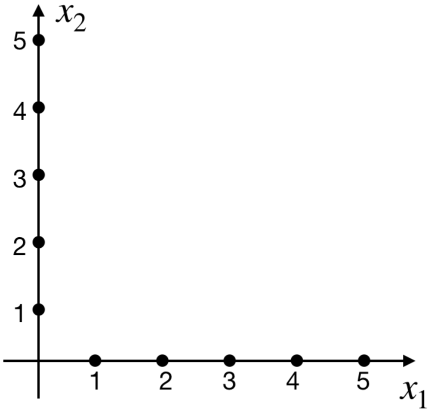
## 4 Decision Boundary (16 points)

Given a classifier that performs classification in $\mathbb{R}^2$ (the space of data points with 2 features $(x_1, x_2)$).

**(a) (6 points)** If the classification rule is as follows:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if } (x_1 \leq 4) \text{ and } (x_2 \geq 4 \text{ and } x_2 \leq 2). \\ 0, & \text{otherwise.} \end{cases}$$

Shade the area on the graph below where the classifier $h(x_1, x_2)$ predicts 1. Make sure you have marked the intercept points on these axes.



**Solution:**

**(b) (10 points)** If the classification rule is as follows:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if} \quad w_1x_1 + w_2x_2 + 8 \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

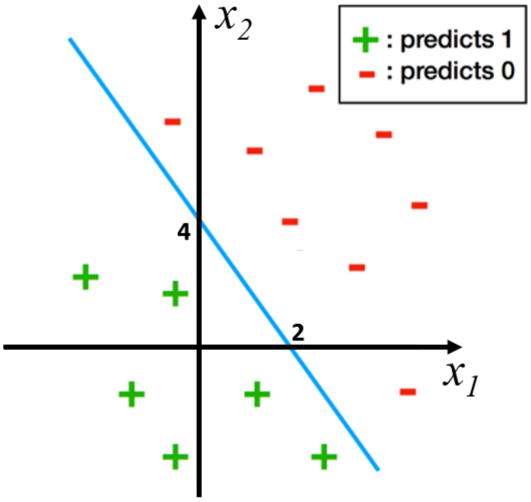


Figure 2: Decision boundary to solve the parameters.

1. Compute the parameters $w_1$, $w_2$ for the decision boundary in Figure 2.
2. Find the predictions for the points A = (1,2), B = (-2,6) and C = (4,-1)

**Solution:**

## 5 Regression Error

Suppose we have a training set of three data points: $S = \{(x_1 = -3, y_1 = 1), (x_2 = 2, y_2 = 3), (x_3 = 0.5, y_3 = 0)\}$. We name these three points $A = (x_1, y_1)$, $B = (x_2, y_2)$, and $C = (x_3, y_3)$.

**(a)** Figure 3 shows a learned regression function $f(x) = x$ from the dataset $S$. Please compute both the absolute error (AE), $|y_i - f(x_i)|$, and squared error (SE), $(y_i - f(x_i))^2$, for the three datapoints $i = 1, 2, 3$ a.k.a. $A$, $B$, and $C$. Then compute the Mean Absolute Error [MAE $= \frac{1}{n}\sum_{i=1}^{n}|y_i - f(x_i)|$], Sum of Squared Errors [SSE $= \sum_{i=1}^{n}(y_i - f(x_i))^2$] and Mean Squared Error [MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$] losses for the training set.

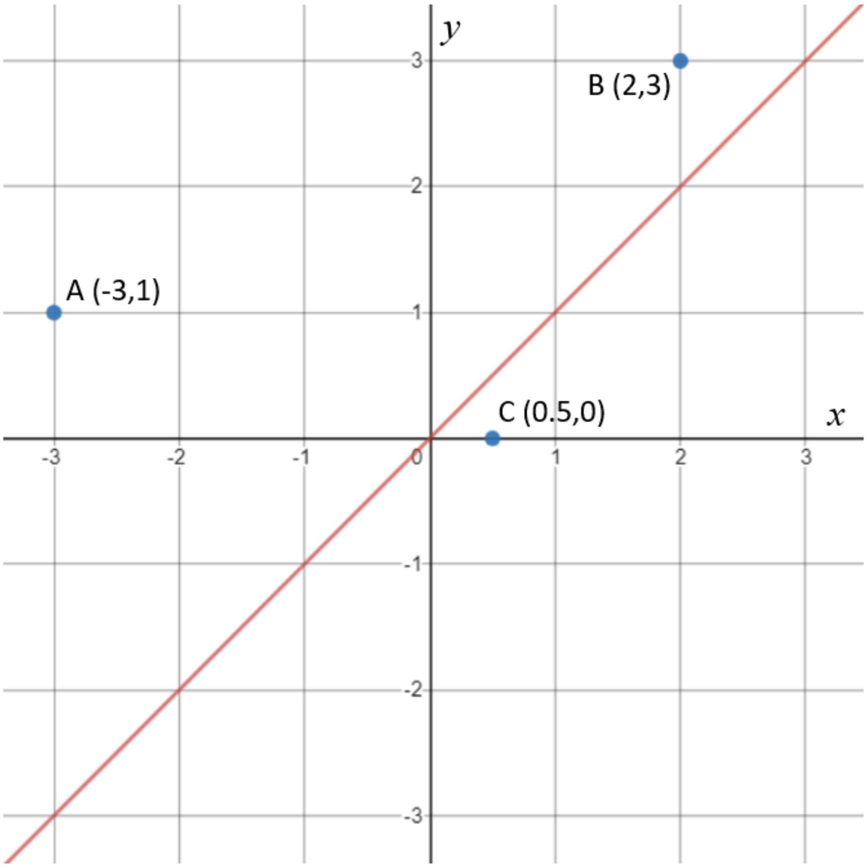


Figure 3: The curve of the function and the data points are given in the figure.

**Solution:**

**(b)** If I told you the regression in figure 3 was fit with the loss function being the L1 norm, could you please write that loss function in terms of $y_i, f(x_i)$ as done in the previous section. Please draw the loss function. Is this loss function convex? Is it everywhere differentiable (i.e., smooth)? Can it be minimized in closed form (analytically)? Why would someone choose to use L1 norm loss function? (1 point each)

Equation of L1 norm: ________________________________________

Convex?: _________

Everywhere differentiable/smooth?: _________

Closed form solution?:________

Why use it?: ________________________________________________

Please draw the shape of the L1 function below:

**Solution:**

## 6 Good Luck on the midterm :)