Gabriel Ferreira
Department of Computer Science
Iowa State University

HW2 — Regression

1 Linear Regression

1.1 Consider the data set $x = \{1, 2, 3, 4, 5, 6, 10\}$, $y = \{0, 1, 3, 2, 20, -6, 80\}$.

a) $\overline{x} = \frac{1}{n}\sum_{1}^{n} xi$

$\overline{x} = \frac{(1+2+3+4+5+6+10)}{7}$
$\overline{x} = 4.43$

$\overline{y} = \frac{1}{n}\sum_{1}^{n} xi$

$\overline{y} = \frac{(0+1+3+2+20-6+80)}{7}$
$\overline{y} = 14.29$

b) $s_x^2 = \frac{1}{n}\sum_{1}^{n}(xi - \overline{x})^2$

$s_x^2 = \frac{53.68}{7}$
$s_x^2 = 7.67$

$s_y^2 = \frac{1}{n}\sum_{1}^{n}(yi - \overline{y})^2$

$s_y^2 = \frac{5421.4}{7}$
$s_y^2 = 774.48$

$s_{xy} = \frac{1}{n}\sum_{1}^{n}(xi - \overline{x}) \times (yi - \overline{y})$

$s_{xy} = \frac{440.12}{7}$
$s_{xy} = 62.88$

c) $y = ax + b$
$a = slope \quad b = intercept$
$a = \frac{s_{xy}}{s_x^2} \quad b = \overline{y} - a\overline{x}$

$a = \frac{62.88}{7.67} \quad b = 14.29 - (8.19 \times 4.43)$
$a = 8.19 \quad b = -22$

The solution is: $y = 8.19x - 22$

d) The predictor (1-1) is expected to outperform (better than) the trivial predictor $\hat{y} = \overline{y}$ because it makes use of the relationship of xy to make predictions. In other hand, $\hat{y} = \overline{y}$ does not make use of x at all, which means it cannot adapt its predictions to any changes in x; instead, it will predict the same value independently of the x input.

e) $y = kx$

Assuming k is the slope, we have the following:

$k = slope$
$k = \frac{s_{xy}}{s_x^2}$

$k = \frac{62.88}{7.67}$
$k = 8.19$

The solution is: $y = 8.19x$

f) The predictor for (1-2) is expected to under perform (worse than) the predictor of the form (1-1), given the fundamental difference between both predictors that is the intercept ((1-2) does not have it).

The intercept is important because it adjust the starting point of the regression line to fit the data better. This can improve the predictor's accuracy as it allows the regression line to be better positioned in relation to the data points, which minimizes the overall error across all x values.

Therefore, the predictor with the intercept is expected to be more flexible and outperform the model that doesn't have it in most scenarios.

g) $y = \alpha x + \beta x^2 + \gamma$

$\theta = (X^TX)^{-1}X^Ty$ where $X = [1, xi, xi^2]$

$$(X^TX) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 10 \\ 1 & 4 & 9 & 16 & 25 & 36 & 100 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 10 & 100 \end{bmatrix} = \begin{bmatrix} 7 & 30 & 191 \\ 31 & 191 & 1441 \\ 191 & 1441 & 12275 \end{bmatrix}$$

$(X^TX)^{-1} = \begin{bmatrix} 7 & 30 & 191 \\ 31 & 191 & 1441 \\ 191 & 1441 & 12275 \end{bmatrix}^{-1} = \begin{bmatrix} 1.52 & -0.60 & -0.05 \\ -0.60 & 0.28 & -0.02 \\ 0.05 & -0.02 & 0.002 \end{bmatrix}$ (Note: This part was calculated in Python)

$$X^Ty = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 10 \\ 1 & 4 & 9 & 16 & 25 & 36 & 100 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 3 \\ 2 \\ 20 \\ -6 \\ 80 \end{bmatrix} = \begin{bmatrix} 100 \\ 883 \\ 8347 \end{bmatrix}$$

$$\theta = (X^TX)^{-1}X^Ty = \begin{bmatrix} 1.52 & -0.60 & -0.05 \\ -0.60 & 0.28 & -0.02 \\ 0.05 & -0.02 & 0.002 \end{bmatrix} \times \begin{bmatrix} 100 \\ 883 \\ 8347 \end{bmatrix} = \begin{bmatrix} 12.418 \\ -9.315 \\ 1.580 \end{bmatrix}$$

The solution is: $y = -9.315x + 1.580x^2 + 12.418$

h) For this comparison, I would say that being better or worse heavily depends on the nature of the dataset we're working with.

The predictor for (1-3) is more flexible, and therefore, more complex given its quadratic nature introducing a second degree term (βx^2) than the predictor of the form (1-1). I would say the predictor (1-3) is expected to be superior than the predictor (1-1) when given a dataset with non-linear relationship or with curves once it will be able to pick up on the parabolic trends.

However, the extra complexity of the predictor (1-3) increases the chances of overfitting the dataset, which isn't a good contribution for being "better than" the predictor (1-1).

Therefore, "better" or "worse" depends on the given datasets and different metrics that can be used. In terms of accuracy, I would say the predictor (1-3) is better. If we were to avoid the risk of overfitting by using a more generalized predictor, then I would use (1-1).

i)

```
def score(y_true, y_pred):

    y_true = np.array(y_true)
    y_pred = np.array(y_pred)

    score = 1 - (((y_true-y_pred)**2).sum() /((y_true-y_true.mean())**2).sum())

    print("The score for this solution is: ")
    return round(score,3)
```

```
def predictor_1_2(X, y):

    x_mean = np.mean(X)
    y_mean = np.mean(y)

    x_variance = np.mean((X - x_mean)**2)
    sxy = np.mean((X.T - x_mean) * (y - y_mean))
    k = sxy / x_variance

    y_pred = k * X

    # Print the model formula
    print("Solution 1-2: ")
    print(f"y = {round(k,2)}x")
    print("")
    return y_pred[:, 0]
```

```
def predictor_1_3(X, y):

    X_design = np.hstack((X**0, X, X**2))

    XTX = np.dot(X_design.T, X_design)
    XTX_inv = inv(XTX)
    XTy = np.dot(X_design.T, y)
    theta = np.dot(XTX_inv, XTy)
    y_pred = np.dot(X_design, theta)

    # Print the model formula
    print("Solution 1-3: ")
    print(f"y = {round(theta[1],3)}x + {round(theta[2], 3)}x^2 + {round(theta[0], 3)}")
    print("")
    return y_pred
```

```
y_pred_1_2 = predictor_1_2(X, Y)
score(Y, y_pred_1_2)
```

Solution 1-2:
$y = 8.19x$

The score for this solution is:
0.04

```
y_pred_1_3 = predictor_1_3(X, Y)
score(Y, y_pred_1_3)
```

Solution 1-3:
$y = -9.315x + 1.58x^2 + 12.419$

The score for this solution is:
0.881

My solution for the functions (1-2) and (1-3) has aligned with my previous answers for them.

j) Mathematically, there is not an unique solution for the regression problem using the given function form and dataset. As explained in class, in order for a solution to be unique, the design matrix X has to be of full rank. For X to be of full rank the number of observations must be grater than or equal to the number of parameters. In our problem, the given function form is a 10th-degree polynomial plus β_0 (11 parameters) and only 7 observations in the given dataset. This means we have more parameters than observations in the dataset, which implicates that X is not full of rank, and therefore, have multiple solutions.

Now, if we fit numpy or sklearn, I am sure they have put programming solutions in place for dealing with scenarios like this. However, even if we get a solution, it should be an solution that will overfit the data and fail to generalize it. Let's fit a polynomial to our X and see its score:

```
from sklearn.preprocessing import PolynomialFeatures
def predictor_poly(X, Y):

    poly = PolynomialFeatures(degree=10)
    X_poly = poly.fit_transform(X)

    model = LinearRegression().fit(X_poly, Y)

    return model

poly = PolynomialFeatures(degree=10)
X_poly = poly.fit_transform(X)
predictor_poly(X, Y).score(X_poly, Y)
```

Outcome: 1.0

2 Logistic Regression

a) I believe this problem should be tackled as regression problem given the fact that people can have 0 to N children. Although we could create labels to tackle this problem as a classification problem, I still think the best approach would be tackling as a regression progress once we're trying to predict quantity.

b) I believe this problem should be tackled as a classification problem. That's because in this problem, we're using image data, which is usually used to predict discrete values. On top of this, each sample can be broken into different classes (2 legs, 4 legs, 6 legs, 8 legs, >8 legs).

c) I believe this problem should be tackled as classification problem as the goal here is to predict whether the person is 0=<80years or 1>80years.

d) I believe this problem should be tackled as regression problem as the goal here is to predict a continuous range of age.

3 A “Bonus" Question (1 pt)

From a scale of 1 to 5, how difficult is HW1? 1 is “I can do it in my sleep". 5 is “Bowen is ridiculous". 0 is “I refuse to answer this question". This is for my own reference to improve the quality of future assignments. Thanks!

4 out of 5.

Submitted by Gabriel Ferreira on .