Multi-Linear Regression

CS109A Introduction to Data Science
Pavlos Protopapas, Natesh Pillai
Lecture Outline

Announcements

Q&A

Part A:

Multi-linear regression

Part B:

Polynomial Regression
Lecture Outline

Announcements

Q&A

Part A:

Multi-linear regression

Part B:

Polynomial Regression
Announcements

• Videos should be available now!
• Deadline for quiz/ex for Monday’s lectures are now on the following Friday @9:30am
• Deadline for quiz/ex for Wednesday’s lectures remain the same: the following Monday @9:30am
• Homework 1 is due tonight @11:59.999999999999999pm
• Homework 2 will be released on today somewhere on planet earth
Lecture Outline

Announcements

Q&A

Part A:
Multi-linear regression

Part B:
Polynomial Regression
How would we choose between a kNN and a linear model to predict our response variable?

Can kNN be used when there are multiple predictor variables?

Does kNN work on binary outcomes?

What would you do if multiple k values are equally good for kNN regression?

Does gradient descent guarantee the same solution as the analytical one?

Can we have some more scenarios on when mean absolute error is useful as a loss function?

How do we make sure that we are not taking a biased subset?

What should the train-test split be?

What is the difference between inference and prediction?
New programmers

Experienced Programmers
Lecture Outline

Announcements

Q&A

Part A:
  Multi-linear regression

Part B:
  Polynomial Regression
Multiple Linear Regression

If you have to guess someone's height, would you rather be told
  • Their weight, only
  • Their weight and sex
  • Their weight, sex, and income
  • Their weight, sex, income, and favorite number

Of course, you'd always want as much data about a person as possible. Even though height and favorite number may not be strongly related, at worst you could just ignore the information on favorite number.

We want our models to be able to take in lots of data as they make their predictions.
Linear Regression in 2D

\[ \hat{f}(x_1) = \hat{\beta}_0 + \hat{\beta}_1 x_1 \]
Linear Regression in 3D

\[ \hat{f}(x_1, x_2) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \]
Multilinear Models

In practice, it is unlikely that any response variable \( Y \) depends solely on one predictor \( x \). Rather, we expect that it is a function of multiple predictors \( f(X_1, \ldots, X_J) \). Using the notation we introduced last lecture,

\[
Y = y_1, \ldots, y_n, \quad X = X_1, \ldots, X_J \quad \text{and} \quad X_j = x_{1j}, \ldots, x_{ij}, \ldots, x_{nj}
\]

In this case, we can still assume a simple form for \( f \) - a multilinear form:

\[
f(X_1, \ldots, X_J) = \beta_0 + \beta_1 X_1 + \cdots + \beta_J X_J
\]

Hence, \( \hat{f} \), has the form:

\[
\hat{f}(X_1, \ldots, X_J) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_J X_J
\]
Response vs. Predictor Variables

This is called X: a.k.a. The Design Matrix

- TV
  - 230.1
  - 44.5
  - 17.2
  - 151.5
  - 180.8

\( n \) observations

\( x_1 \)

y: The response variable

- sales
  - 22.1
  - 10.4
  - 9.3
  - 18.5
  - 12.9
Response vs. Predictor Variables

This is called $X$: a.k.a. The Design Matrix

<table>
<thead>
<tr>
<th></th>
<th>TV</th>
<th>radio</th>
<th>newspaper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>230.1</td>
<td>37.8</td>
<td>69.2</td>
</tr>
<tr>
<td>2</td>
<td>44.5</td>
<td>39.3</td>
<td>45.1</td>
</tr>
<tr>
<td>3</td>
<td>17.2</td>
<td>45.9</td>
<td>69.3</td>
</tr>
<tr>
<td>4</td>
<td>151.5</td>
<td>41.3</td>
<td>58.5</td>
</tr>
<tr>
<td>5</td>
<td>180.8</td>
<td>10.8</td>
<td>58.4</td>
</tr>
</tbody>
</table>

$y$: The response variable

<table>
<thead>
<tr>
<th>sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.1</td>
</tr>
<tr>
<td>10.4</td>
</tr>
<tr>
<td>9.3</td>
</tr>
<tr>
<td>18.5</td>
</tr>
<tr>
<td>12.9</td>
</tr>
</tbody>
</table>
Multilinear Model, example

For our data

\[ Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper \]

In linear algebra notation

\[
Y = \begin{pmatrix} Sales_1 \\ \vdots \\ Sales_n \end{pmatrix},
X = \begin{pmatrix} 1 & TV_1 & Radio_1 & News_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & TV_n & Radio_n & News_n \end{pmatrix},
\beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}
\]

\[
Sales_1 = \begin{pmatrix} 1 & TV_1 & Radio_1 & News_1 \end{pmatrix} \times \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}
\]
Multilinear Model, example

\[
Sales_1 = \begin{bmatrix} 1 & TV_1 & Radio_1 & News_1 \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{bmatrix}
\]

\[
Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper
\]

\[
Y = X\beta
\]
Multiple Linear Regression

Given a set of observations,

\[
\{(x_{11}, x_{12}, y_1), (x_{21}, x_{22}, y_2), \ldots, (x_{n1}, x_{n2}, y_n)\}
\]

the data and the model can be expressed in vector notation,

\[
Y = X\beta
\]

where,

\[
\begin{align*}
Y &= \begin{pmatrix} y_1 \\ y_2 \\ \vdots \end{pmatrix} \\
X &= \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \end{pmatrix} \\
\beta &= \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}
\end{align*}
\]
RECAP: Transpose of a matrix

You can perform transpose over numpy objects by calling `np.transpose()` or `ndarray.T`
RECAP: Inverse of a matrix

When we multiply a number by its reciprocal we get 1.

\[ \frac{1}{n} \neq 1 \]

When we multiply a matrix by its inverse, we get the Identity Matrix

\[ A A^{-1} = I \]

In [16]: x = np.array([[1,2],[3,4]])
    
    ...:
    ...: #Inverse array x
    ...: invX = np.linalg.inv(x)
    ...:
    ...: print(invX)
    ...:
    ...: #Verifying
    ...: print(np.dot(x, invX))

\[
\begin{bmatrix}
-2. & 1. \\
1.5 & -0.5
\end{bmatrix}
\]

\[
\begin{bmatrix}
1.00000000e+00 & 1.11022302e-16 \\
0.00000000e+00 & 1.00000000e+00
\end{bmatrix}
\]

`numpy.linalg.inv()` is used to calculate the inverse of a matrix (if it exists!)
Multiple Linear Regression

The model takes a simple algebraic form:

$$Y = X\beta$$

We will again choose the MSE as our loss function, which can be expressed in vector notation as

$$MSE(\beta) = \frac{1}{n} \| Y - X\beta \|^2$$

$$MSE(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2$$
\[ \text{MSE} (\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_{i1} \beta_1 - x_{i2} \beta_2)^2 \]

\[ \text{MSE} (\beta) = (y_1 - x_{11} \beta_1 - x_{12} \beta_2)^2 + (y_2 - x_{21} \beta_1 - x_{22} \beta_2)^2 + \ldots \]
\[ \text{MSE}(\beta) = (y_1 - x_{11}\beta_1 - x_{12}\beta_2)^2 + \\
(y_2 - x_{21}\beta_1 - x_{22}\beta_2)^2 + \ldots \]

\[
\frac{\partial L}{\partial \beta_1} = -2x_{11}(y_1 - x_{11}\beta_1 - x_{12}\beta_2) \\
-2x_{21}(y_2 - x_{21}\beta_1 - x_{22}\beta_2) - \ldots \\
\frac{\partial L}{\partial \beta_2} = -2x_{12}(y_1 - x_{11}\beta_1 - x_{12}\beta_2) \\
-2x_{22}(y_2 - x_{21}\beta_1 - x_{22}\beta_2) - \ldots 
\]
\[
\left( \frac{\partial L}{\partial \beta_1} \right) = \begin{pmatrix}
-2x_{11}(y_1 - x_{11}\beta_1 - x_{12}\beta_2) \\
-2x_{12}(y_1 - x_{11}\beta_1 - x_{12}\beta_2)
\end{pmatrix} - 2x_{21}(y_2 - x_{21}\beta_1 - x_{22}\beta_2) - \ldots
\]

\[
\left( \frac{\partial L}{\partial \beta_2} \right) = \begin{pmatrix}
-2x_{11}(y_1 - x_{11}\beta_1 - x_{12}\beta_2) \\
-2x_{12}(y_1 - x_{11}\beta_1 - x_{12}\beta_2)
\end{pmatrix} - 2x_{22}(y_2 - x_{21}\beta_1 - x_{22}\beta_2) - \ldots
\]

\[
\left( \frac{\partial L}{\partial \beta_1} \right) = -2 \begin{pmatrix}
x_{11} & x_{21} & \ldots \\
x_{12} & x_{22} & \ldots
\end{pmatrix} \begin{pmatrix}
y_1 - x_{11}\beta_1 - x_{12}\beta_2 \\
y_2 - x_{21}\beta_1 - x_{22}\beta_2 \\
\vdots
\end{pmatrix}
\]
\[
\left( \frac{\partial L}{\partial \beta_1} \right) = -2 \begin{pmatrix} x_{11} & x_{21} & \cdots \\ x_{12} & x_{22} & \cdots \end{pmatrix} \begin{pmatrix} y_1 - x_{11} \beta_1 - x_{12} \beta_2 \\ y_2 - x_{21} \beta_1 - x_{22} \beta_2 \\ \cdots \end{pmatrix}
\]

\[
\left( \frac{\partial L}{\partial \beta_2} \right) = -2 \begin{pmatrix} x_{11} & x_{21} & \cdots \\ x_{12} & x_{22} & \cdots \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \cdots \end{pmatrix} - \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \cdots \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}
\]
\[
\begin{pmatrix}
\frac{\partial L}{\partial \beta_1} \\
\frac{\partial L}{\partial \beta_2}
\end{pmatrix}
= -2 \begin{pmatrix}
x_{11} & x_{21} & \ldots \\
x_{12} & x_{22} & \ldots 
\end{pmatrix}
\left[
\begin{pmatrix}
y_1 \\
y_2 \\
\ldots
\end{pmatrix}
- \begin{pmatrix}
x_{11} & x_{12} \\
x_{21} & x_{22} \\
\ldots & \ldots
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2
\end{pmatrix}
\right]
\]

\[
\begin{pmatrix}
\frac{\partial L}{\partial \beta}
\end{pmatrix}
= -2X^T (Y - X\beta)
\]

For optimization, we set the values of the partial derivative to zero, i.e.,

\[
\begin{pmatrix}
\frac{\partial L}{\partial \beta}
\end{pmatrix}
= 0
\]
\[
\begin{pmatrix}
\frac{\partial L}{\partial \beta_1} \\
\frac{\partial L}{\partial \beta_2}
\end{pmatrix}
= -2
\begin{pmatrix}
x_{11} & x_{21} & \ldots \\
x_{12} & x_{22} & \ldots 
\end{pmatrix}
\begin{pmatrix}
y_1 \\
y_2 \\
\vdots
\end{pmatrix}
- \begin{pmatrix}
x_{11} & x_{12} \\
x_{21} & x_{22} \\
\vdots & \vdots 
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2
\end{pmatrix}
\]

\[
\frac{\partial L}{\partial \beta} = 0 \quad \Rightarrow \quad -2X^T(Y - X\beta) = 0
\]

\[
\Rightarrow \quad X^T(Y - X\beta) = 0
\]
Optimization

\[ X^T (Y - X\beta) = 0 \]

Which gives us,

\[ X^T X \beta = X^T Y \]

Multiplying on both sides with \((X^T X)^{-1}\)

\[ (X^T X)^{-1}X^T X \beta = (X^T X)^{-1}X^T Y \]

\[ \Rightarrow \beta = (X^T X)^{-1}X^T Y \]
RECAP: Multiple Linear Regression

The model takes a simple algebraic form: \( \mathbf{Y} = \mathbf{X}\beta + \epsilon \)

We will again choose the \textbf{MSE} as our loss function, which can be expressed in vector notation as

\[
\text{MSE}(\beta) = \frac{1}{n} \| \mathbf{Y} - \mathbf{X}\beta \|^2
\]

Minimizing the MSE using vector calculus yields,

\[
\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} = \arg\min_{\beta} \text{MSE}(\beta).
\]
Multiple Linear Regression

\[ \hat{\beta} = \left( X^\top X \right)^{-1} X^\top Y = \arg\min_{\beta} \text{MSE}(\beta). \]

```python
>>> import numpy as np
>>> X = ...
>>> y = ...
>>> X_sq = X.T @ X
>>> X_inv = np.linalg.inv(X.T @ X)
>>> beta_hat = X_inv @ (X.T @ y)
```
Digestion Time
Interpreting multi-linear regression

For linear models, it is easy to interpret the model parameters.

When we have a large number of predictors: $X_1, ..., X_J$, there will be a large number of model parameters, $\beta_1, \beta_2, ..., \beta_J$.

Looking at the values of $\beta$’s is impractical, so we visualize these values in a **feature importance** graph.

The feature importance graph shows which predictors has the **most impact** on the model’s prediction.
Collinearity and multicollinearity refers to the case in which two or more predictors are correlated (related).

The regression coefficients are not uniquely determined. In turn it hurts the interpretability of the model as then the regression coefficients are not unique and have influences from other features.

Both limit and rating have positive coefficients, but it is hard to understand if the balance is higher because of the rating or is it because of the limit? If we remove limit then we achieve almost the same model performance, but the coefficients change.
Collinearity refers to the case in which two or more predictors are correlated (related).

We will re-visit collinearity in the next lecture when we address overfitting, but for now we want to examine how does collinearity affects our confidence on the coefficients and consequently on the importance of those coefficients.
Qualitative Predictors

So far, we have assumed that all variables are quantitative. But in practice, often some predictors are **qualitative**.

**Example:** The *credit data set* contains information about balance, age, cards, education, income, limit, and rating for a number of potential customers.

<table>
<thead>
<tr>
<th>Income</th>
<th>Limit</th>
<th>Rating</th>
<th>Cards</th>
<th>Age</th>
<th>Education</th>
<th>Sex</th>
<th>Student</th>
<th>Married</th>
<th>Ethnicity</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.890</td>
<td>3606</td>
<td>283</td>
<td>2</td>
<td>34</td>
<td>11</td>
<td>Male</td>
<td>No</td>
<td>Yes</td>
<td>Caucasian</td>
<td>333</td>
</tr>
<tr>
<td>106.02</td>
<td>6645</td>
<td>483</td>
<td>3</td>
<td>82</td>
<td>15</td>
<td>Female</td>
<td>Yes</td>
<td>Yes</td>
<td>Asian</td>
<td>903</td>
</tr>
<tr>
<td>104.59</td>
<td>7075</td>
<td>514</td>
<td>4</td>
<td>71</td>
<td>11</td>
<td>Male</td>
<td>No</td>
<td>No</td>
<td>Asian</td>
<td>580</td>
</tr>
<tr>
<td>148.92</td>
<td>9504</td>
<td>681</td>
<td>3</td>
<td>36</td>
<td>11</td>
<td>Female</td>
<td>No</td>
<td>No</td>
<td>Asian</td>
<td>964</td>
</tr>
<tr>
<td>55.882</td>
<td>4897</td>
<td>357</td>
<td>2</td>
<td>68</td>
<td>16</td>
<td>Male</td>
<td>No</td>
<td>Yes</td>
<td>Caucasian</td>
<td>331</td>
</tr>
</tbody>
</table>
Qualitative Predictors

If the predictor takes only two values, then we create an **indicator** or **dummy variable** that takes on two possible numerical values. For example, for the sex, we create a new variable:

\[ x_i = \begin{cases} 
1 & \text{if } i \text{ th person is female} \\
0 & \text{if } i \text{ th person is male} 
\end{cases} \]

We then use this variable as a predictor in the regression equation.

\[ y_i = \beta_0 + \beta_1 x_i = \begin{cases} 
\beta_0 + \beta_1 & \text{if } i \text{ th person is female} \\
\beta_0 & \text{if } i \text{ th person is male} 
\end{cases} \]
Question: What is interpretation of $\beta_0$ and $\beta_1$?
Qualitative Predictors

Question: What is interpretation of $\beta_0$ and $\beta_1$?

- $\beta_0$ is the **average** credit card balance among **males**,

- $\beta_0 + \beta_1$ is the **average** credit card balance among **females**,

- and $\beta_1$ the average **difference** in credit card balance between **females** and **males**.

Example: Calculate $\beta_0$ and $\beta_1$ for the Credit data.
You should find $\beta_0 \sim \$509$, $\beta_1 \sim \$19$
More than two levels: One hot encoding

Often, the qualitative predictor takes more than two values (e.g. ethnicity in the credit data).

In this situation, a single dummy variable cannot represent all possible values.

We create additional dummy variable as:

\[ x_{i,1} = \begin{cases} 
1 & \text{if } i \text{ th person is Asian} \\
0 & \text{if } i \text{ th person is not Asian} 
\end{cases} \]

\[ x_{i,2} = \begin{cases} 
1 & \text{if } i \text{ th person is Caucasian} \\
0 & \text{if } i \text{ th person is not Caucasian} 
\end{cases} \]
More than two levels: One hot encoding

We then use these variables as predictors, the regression equation becomes:

\[ y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} = \begin{cases} 
\beta_0 + \beta_1 & \text{if } i \text{ th person is Asian} \\
\beta_0 + \beta_2 & \text{if } i \text{ th person is Caucasian} \\
\beta_0 & \text{if } i \text{ th person is African American} 
\end{cases} \]

**Question:** What is the interpretation of \( \beta_0, \beta_1, \beta_2 \)?
Beyond linearity

So far, we assumed:

- linear relationship between $X$ and $Y$
- the residuals $r_i = y_i - \hat{y}_i$ were uncorrelated (taking the average of the square residuals to calculate the MSE implicitly assumed uncorrelated residuals)

These assumptions need to be verified using the data. This is often done by **visually inspecting the residuals**.
Residual Analysis

Linear assumption is correct. There is no obvious relationship between residuals and $x$. Histogram of residuals is symmetric and normally distributed.

Linear assumption is incorrect. There is an obvious relationship between residuals and $x$. Histogram of residuals is symmetric but not normally distributed.

Note: For multi-regression, we plot the residuals vs predicted $y$, $\hat{y}$, since there are too many $x$'s and that could wash out the relationship.
Beyond linearity: **synergy effect** or **interaction effect**

We also assume that the average effect on sales of a one-unit increase in TV, is always $\beta_1$ regardless of the amount spent on radio or newspaper.

**Synergy effect** or **interaction effect** states that when an increase on the radio budget affects the effectiveness of the TV spending on sales.

To account for it, we simply add a term as:

\[
Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2
\]
What does it mean? No interaction term

\[ balance = \beta_0 + \beta_1 \times \text{income} + \beta_2 \times \text{student} \]

\[ \text{student} = \begin{cases} 0 & balance = \beta_0 + \beta_1 \times \text{income} \\ 1 & balance = \beta_0 + \beta_1 \times \text{income} + \beta_2 \end{cases} \rightarrow balance = (\beta_0 + \beta_2) + \beta_1 \times \text{income} \]
What does it mean? With interaction term

\[ balance = \beta_0 + \beta_1 \times \text{income} + \beta_2 \times \text{student} + \beta_3 \times \text{income} \times \text{student} \]

\[
\begin{align*}
\text{student} &= \begin{cases} 
0 & \text{balance} = \beta_0 + \beta_1 \times \text{income} \\
1 & \text{balance} = \beta_0 + \beta_1 \times \text{income} + \beta_2 + \beta_3 \times \text{income}
\end{cases}
\]

\[ \rightarrow balance = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \text{income} \]
Digestion Time
Too many predictors, collinearity and too many interaction terms leads to **OVERFITTING!**
Exercise: Simple Multi-linear Regression

The aim of this exercise is to understand how to use multi regression. Here we will observe the difference in MSE for each model as the predictors change.

<table>
<thead>
<tr>
<th>TV</th>
<th>Radio</th>
<th>Newspaper</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>230.1</td>
<td>37.8</td>
<td>69.2</td>
</tr>
<tr>
<td>1</td>
<td>44.5</td>
<td>39.3</td>
<td>45.1</td>
</tr>
<tr>
<td>2</td>
<td>17.2</td>
<td>45.9</td>
<td>69.3</td>
</tr>
<tr>
<td>3</td>
<td>151.5</td>
<td>41.3</td>
<td>58.5</td>
</tr>
<tr>
<td>4</td>
<td>180.8</td>
<td>10.8</td>
<td>58.4</td>
</tr>
</tbody>
</table>

Instructions:

- Read the file Advertisement.csv as a dataframe.
- For each instance of the predictor combination, form a model. For example, if you have 2 predictors, A and B, you will end up getting 3 models - one with only A, one with only B, and one with both A and B.
- Split the data into train and test sets.
- Compute the MSE of each model.
- Print the Predictor - MSE value pair.

Hints: