Exercises on regression

Data

The data set we will use in this set of exercises is a made-up data of short messages, e.g., tweets. The data summarizes a hypothetical twitter corpus where we have calculated length of the sentences (`sent_len`) and average token length per message (`word_len`). Given these two variables, we want to predict the age of the author (`age`) in this exercise set.

You can download the data from URL http://coltekin.net/cagri/courses/ml/data/fake-twitter-data.csv, use directly to load to the environment you are working.

Exercises

1. Fit a regression model predicting `age` only from `sent_len`.
   - What is the intercept and slope?
   - How do you interpret the coefficients?
   - What is the coefficient of determination (R^2) and what does it mean?
2. Fit another model, this time, including `word_len` as a second predictor.
   - Did the model fit improve?
   - Which predictor is more important?
3. Find the predicted `age` values by the model fit in exercise 2 for the following combinations of word and sentence lengths.

`word_len`	`sent_len`
2	10
5	50
10	1
10	100
4	70

4. Plot the relevant data points, and the regression line you found in exercise 1.

Tips

0. Loading the data
   - R:

```
d <- read.csv('http://coltekin.net/cagri/courses/ml/data/fake-twitter-data.csv')
```

- Python:

```
import pandas as pd
d = pd.read_csv('http://coltekin.net/cagri/courses/ml/data/fake-twitter-data.csv')
```

1. To fit a regression model, check the coefficients, R^2

   - In R

   ```
   m <- lm(age ~ sent_len, data=d)
   summary(m)
   ```

   - In Python (with sklearn)

   ```
   from sklearn.linear_model import LinearRegression
   m1 = LinearRegression()
   m1.fit(d[['sent_len']], d['word_len'])
   m1.coef_
   m1.intercept_
   m1.score()
   ```

2. Multiple predictors

   - In R

   ```
   m2 <- lm(age ~ sent_len + word_len, data=d)
   ```

   - In Python

   ```
   m2 = LinearRegression()
   m2.fit(d[['sent_len', 'word_len']], d['word_len'])
   ```

3. Obtaining predictions

   - In R:

   ```
   predict(m2, newdata=data.frame(sent_len=c(10, 50, 1, 100, 70),
           word_len=c(2,5,10,10,4))
   ```

   - In Python:

   ```
   import numpy as np
   m2.predict(np.matrix('10, 2; 50, 5; 1, 10; 100, 10; 70, 4'))
   ```

4. Plotting

   - In R:

   ```
   plot(age ~ sent_len, data=d)
   abline(m2)
   ```

   - In Python:

```
import matplotlib.pyplot as plt
plt.scatter(d.sent_len, d.age)
xmin = min(d.sent_len)
xmax = max(d.sent_len)
ymin, ymax = m1.predict(np.matrix([[xmin],[xmax]]))
plt.plot((xmin, xmax), (ymin, ymax), color='red', linewidth=2)
plt.show()
```