Linear Regression

September 27, 2019

```
In [1]: import numpy as np
```

1 Linear Regression

```
In [2]: x = np.random.random(size=(5, 5))
        y = np.random.random(size=5)
```

1.1 Least-squares method of fitting

The idea is the minimize the residual sum of squares:

$$\text{RSS}(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2$$

We can write that in matrix form as follows:

$$\text{RSS}(\beta) = (y - X\beta)^T(y - X\beta)$$

where X is an $N \times p$ matrix with each row representing an input vector and y is a an N-vector of the outputs in the training set

Differentiating with respect to β, we get

$$\frac{\partial}{\partial\beta}\text{RSS}(\beta) = X^T(y - X\beta) = 0$$

If X^TX is nonsingular, that is, if it has an inverse (i.e. determinant is non zero), then

$$\hat{\beta} = (X^TX)^{-1}X^Ty$$

```
In [3]: def least_squares(X, y):
            N = X.shape[0]
            p = X.shape[1]

            assert(y.shape[0] == N)

            beta = np.linalg.inv(X.T.dot(X)).dot(X.T.dot(y))

            return beta
```

```
In [4]: assert(np.allclose(least_squares(x, y),
                           np.linalg.lstsq(x, y, rcond=None)[0]))
```