Multiple Linear Regression

Introduction

In the previous notes, we only have one independent variable or one feature. In most cases of machine learning, we want to include more than one feature or we want to have a hypothesis that is not simply a straight line. For the first example, we may want to consider not only the floor area but also the storey level to predict the resale price of HDB houses. For the second example, we may want to model the relationship not as a straight line but rather as quadratic. Can we still use linear regression to do these?

This section discusses how we can include more than one feature and how to model our equation beyond a simple straight line using multiple linear regression.

Hypothesis

Recall that in linear regression, our hypothesis is written as follows.
$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$$

where x is the only independent variable or feature. In multiple linear regression, we have more than one feature. We will write our hypothesis as follows.
$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_n x_n$$

In the above hypothesis, we have n features. Note also that we can assume to have $x_0 = 1$ with $\hat{\beta}_0$ as its coefficient.

We can write this in terms of a row vector, where the features are written as
$$\mathbf{X} = \begin{bmatrix} x_0 & x_1 & \ldots & x_n \end{bmatrix} \in {\rm I\!R}^{n+1}$$

Note that the dimension of the feature is $n+1$ because we have $x_0 = 1$ which is a constant of 1.

The parameters can be written as follows.
$$\mathbf{\hat{b}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \ldots \\ \hat{\beta}_n \end{bmatrix} \in {\rm I\!R}^{n+1}$$

Our system equations for all the data points can now be written as follows.
$$\hat{y}(x^1) = \hat{\beta}_0 + \hat{\beta}_1 x_1^1 + \hat{\beta}_2 x_2^1 + \ldots + \hat{\beta}_n x_n^1$$$$\hat{y}(x^2) = \hat{\beta}_0 + \hat{\beta}_1 x_1^2 + \hat{\beta}_2 x_2^2 + \ldots + \hat{\beta}_n x_n^2$$$$\ldots$$$$\hat{y}(x^m) = \hat{\beta}_0 + \hat{\beta}_1 x_1^m + \hat{\beta}_2 x_2^m + \ldots + \hat{\beta}_n x_n^m$$

In the above equations, the superscript indicate the index for the data points from 1 to m, assuming there are m data points.

To write the hypothesis as a matrix equation we first need to write the features as a matrix for all the data points.
$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & \ldots & x_n^1 \\ 1 & x_1^2 & \ldots & x_n^2 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_1^m & \ldots & x_n^m \end{bmatrix} \in {\rm I\!R}^{m \times (n+1)}$$

with this, we can now write the hypothesis as a matrix multiplication.
$$\mathbf{\hat{y}} = \mathbf{X} \times \mathbf{\hat{b}}$$

Notice that this is the same matrix equation as a simple linear regression. What differs is that $\mathbf{\hat{b}}$ contains more than two parameters. Similarly, the matrix $\mathbf{X}$ is now of dimension $m\times(n+1)$ where m is the number of data points and $n+1$ is the number of parameters. Next, let's see how we can calculate the cost function.

Cost Function

Recall that the cost function is written as follows.
$$J(\hat{\beta}_0, \hat{\beta}_1) = \frac{1}{2m}\Sigma^m_{i=1}\left(\hat{y}(x^i)-y^i\right)^2$$

We can rewrite the square as a multiplication instead and make use of matrix multplication to express it.
$$J(\hat{\beta}_0, \hat{\beta}_1) = \frac{1}{2m}\Sigma^m_{i=1}\left(\hat{y}(x^i)-y^i\right)\times \left(\hat{y}(x^i)-y^i\right)$$

Writing it as matrix multiplication gives us the following.
$$J(\hat{\beta}_0, \hat{\beta}_1) = \frac{1}{2m}(\mathbf{\hat{y}}-\mathbf{y})^T\times (\mathbf{\hat{y}}-\mathbf{y})$$

$$\hat{\beta}_0 \hat{\beta}_1 = \frac{1}{2m}(\mathbf{\hat{y}} - \mathbf{y})^T \times (\mathbf{\hat{y}} - \mathbf{y})$$

This equation is exactly the same as the simple linear regression.

Gradient Descent

Recall that the update function for gradient descern algorithm for a linear regression is given as follows.
$$\hat{\beta}_0 = \hat{\beta}_0 - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)$$$$\hat{\beta}_1 = \hat{\beta}_1 - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)x^i$$

In the case of multiple linear regression, we have more than one feature and so we need to differentiate for each θ_j. Doing this will result in a system of equation as follows.
$$\hat{\beta}_0 = \hat{\beta}_0 - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)x_0^i$$$$\hat{\beta}_1 = \hat{\beta}_1 - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)x_1^i$$$$\hat{\beta}_2 = \hat{\beta}_2 - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)x_2^i$$$$\ldots$$$$\hat{\beta}_n = \hat{\beta}_n - \alpha \frac{1}{m}\Sigma_{i=1}^m\left(\hat{y}(x^i) - y^i\right)x_n^i$$

Note that $x_0 = 1$ for all i.

We can now write the gradient descent update function using matrix operations.
$$\mathbf{\hat{b}} = \mathbf{\hat{b}} - \alpha\frac{1}{m} \mathbf{X}^T \times (\mathbf{\hat{y}} - y)$$

Substituting the equation for $\mathbf{\hat{y}}$ gives us the following.
$$\mathbf{\hat{b}} = \mathbf{\hat{b}} - \alpha\frac{1}{m} \mathbf{X}^T \times (\mathbf{X}\times \mathbf{\hat{b}} - y)$$

Again, this is exactly the same as the simple linear regression.

This means that all our equations have not changed and what we need to do is create the right parameter vector $\mathbf{\hat{b}}$ and the matrix $\mathbf{X}$. Once we constructed these vector and matrix, all the other equations remain the same.

Polynomial Model

There are time that even when there is only one feature we may want to have a hypothesis that is not a straight line. An example of would be if our model is a quadratic equation. We can use multiple linear regression to create hypothesis beyond a straight line.

Recall that in multiple linear regression, the hypothesis is writen as follows.
$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_n x_n$$

To have a quadratic hypothesis, we can set the following:
$$x_1 = x$$$$x_2 = x^2$$

And so, the whole equation can be written as
$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 $$

In this case, the matrix for the features becomes as follows.
$$\mathbf{X} = \begin{bmatrix} 1 & x^{(1)} & (x^2)^{(1)} \\ 1 & x^{(2)} & (x^2)^{(2)} \\ \ldots & \ldots \\ 1 & x^{(3)} & (x^2)^{(m)} \end{bmatrix} \in {\rm I\!R}^{m \times 3}$$

In the notation above, we have put the index for the data point inside a bracket to avoid confusion with the power.

We can genearalize this to any power of polynomial where each power is treated as each feature in the matrix. This means that if we want to to model the data using any other polynomial equation, what we need to do is to transform the $\mathbf{X}$ matrix in such a way that each column in $\mathbf{X}$ represents the right degree of polynomial. Column zero is for x^0, column one is for x^1, column two is for x^2, and similarly all the other columns until we have column n for x^n.

The parameters can be found using the same gradient descent that minimizes the cost function.