linear regression with multiple variables

blue dots: training examples

plane: hypothesis

n = number of features

m = number of examples

training examples: (the rows are the examples)

hypothesis function

(a column vector of size n + 1)

(a row vector of size n + 1)

To be able to use a vectorized implementation, we add a column of 1s before X.

1 \* (n + 1) (n + 1) \* 1 1 \* 1

cost function

This function takes in the parameters as the inputs and output the ‘cost’, or the effectiveness of the hypothesis using the input parameters by calculating the average squared sum error of each training example.

gradient descent

α = learning rate

This iterative algorithm takes small steps at a time with the direction calculated by the partial derivatives multiplied by the scalar alpha towards a local optimum so that output calculated by the hypothesis function using the parameters will be the closest to the given output.

Remember that the thetas need to be updated simultaneously.

After calculating the partial derivatives:

we can replace second line with: so we can have a vectorized implementation.

returns a column vector of predicted values of size m.

(n + 1) \* m m \* 1

(n + 1) \* 1