

1. **Notation**
   * : number of training examples
   * : number of features
   * : input variable
   * : output variable
   * : input features of ith training example
   * : value of feature j in ith training example
   * : output of ith training example
   * (: ith training example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | : size | : # of bedrooms | : # of floors | : age of house |  |
| 1 | 2104 | 5 | 1 | 45 | 460 |
| 1 | 1416 | 3 | 2 | 40 | 232 |

* + This example shows a training data with:
    - 2 training examples
    - 4 features (by convention, there will always be feature 0 & it will always be 1)
    - : feature 3 of the second training example, which is 2

1. **Hypothesis**
   * Multivariate linear regression equation:
     + Note: we can write as following:
   * Polynomial regression equation (some examples):
2. **Cost Functions**
   * Squared error (SE) cost function:
3. **Gradient Descent Algorithm Using SE Loss Function for Multivariate Regression**

Repeat until convergence {

}

* + Note:

1. **Feature Scaling**
   * Make sure that the features are on similar scale (i.e. Data normalization)
   * Feature scaling helps making the contour plot of the cost function less skewed. Thus, helping the gradient descent move through a much direct path
   * Here are some formulas used for features scaling:
   * Note: we don’t apply feature scaling to feature 0 (i.e. ) because by convention, it will always be 1
2. **Normal Equation**
   * A method to solve for analytically
   * We can apply the following formula to solve for :
   * Example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | : size | : # of bedrooms | : # of floors | : age of house |  |
| 1 | 2104 | 5 | 1 | 45 | 460 |
| 1 | 1416 | 3 | 2 | 40 | 232 |
| 1 | 1534 | 3 | 2 | 30 | 315 |
| 1 | 852 | 2 | 1 | 36 | 178 |

Construct a matrix and from the data:

From here, we can solve for …

* + Note:
    - Using normal equation method does not require us to do feature scaling.
    - There are some pros and cons of normal equation method

|  |  |
| --- | --- |
| **Gradient descent** | **Normal equation** |
| * Need to choose | * No need to choose |
| * Need many iterations | * Don’t need to iterate |
| * Works well even when n is large | * Slow when n is large because it needs to compute the inverse of (~ |
|  | * may not be invertible |

* + - If the matrix is not invertible, we can try:
      * Pseudoinverse the matrix (?)
      * Remove redundant features (linearly dependent features)
      * Have fewer features (Could affect the correctness)

1. **Note**
   * In multivariate regression, two problems may arise:
     + Overfitting: caused by adding too many independent variables; they account for more variance but add nothing to the model
     + Multicollinearity: happens when some/all of the independent variables () are correlated with each other 🡪 we want to eliminate multicollinearity.