

Simple linear regression model

1. **Notation:**
   * : number of training examples
   * : input variable
   * : output variable
   * : input of ith training example
   * : output of ith training example
   * (: ith training example
2. **Hypothesis:**
3. **Cost (Loss) Functions:** 
   * Squared error (SE) cost function:
   * Sum squared error (SSE) cost function:
   * Mean squared error (MSE) cost function:
   * Root mean squared error (RMSE) cost function:
     + For classification problem, RMSE is not a good choice for loss function
4. **General Gradient Descent Algorithm for Simple Linear Regression:**

Repeat until convergence {

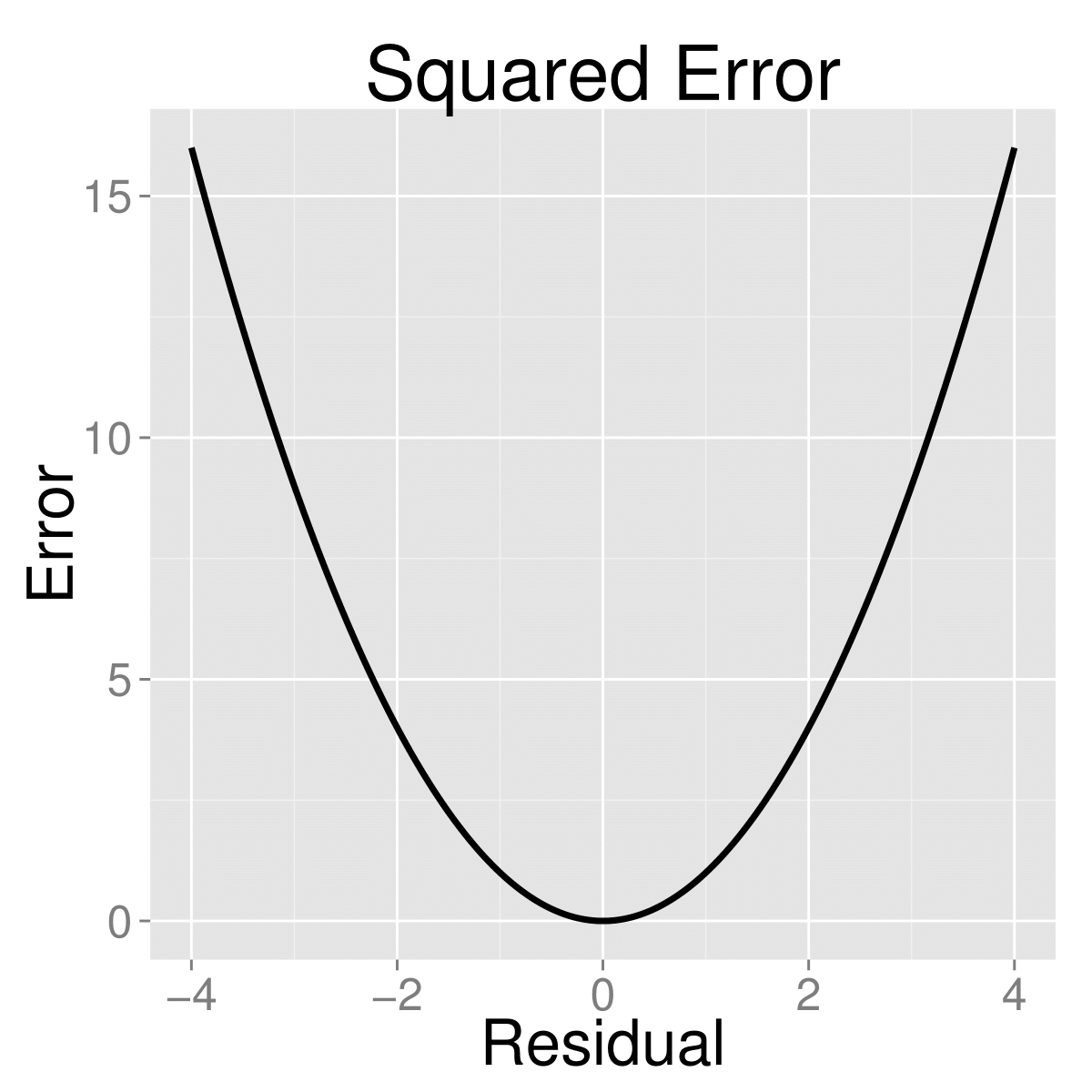
}

1. **Gradient Descent Algorithm Using SE Loss Function in SLR:**

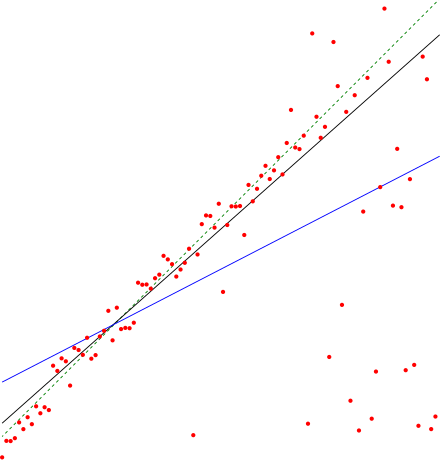
Repeat until convergence {

}

1. **Note:**
   * We have a cost function and we want to minimize the cost function to obtain an optimal solution for the hypothesis function 🡪 Use gradient descent algorithm to continuously finding values that are closest to the
   * The value is called: learning rate.
     + It determines the aggressiveness of the learning algorithm
     + If is too small 🡪 gradient descent can be slow
     + If is too big 🡪 gradient descent can overshoot the minimum. It may fail to converge, or even diverge



Example of the squared-error loss function



Example of different hypothesis base on different set of

* + For SLR, when we plot the cost function, the resulting graph is a convex function (has only 1 global minimum). Because of this, another approach from using gradient descent is to set the partial derivative with respect to to be 0 in order to find the minimum.
  + Another approach to find the hypothesis function would be using Pearson’s R correlation coefficient (PCC):
  + Before deciding whether we should apply SLR or not, if we compute the Pearson’s R correlation coefficient and find that , then we can say that a linear relationship exists