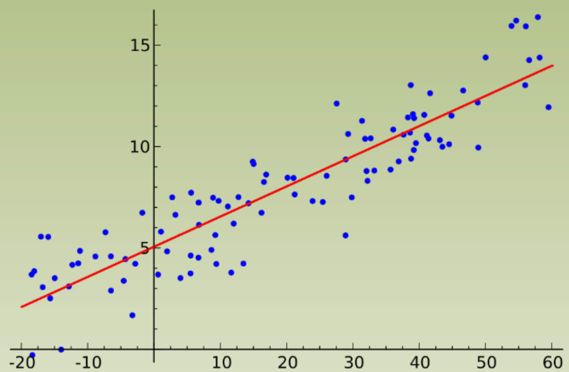
Linear Regression with Gradient Descent

Linear Regression Model



- $f(x) = \mathrm{w}_0 + \mathrm{w}_1 x$
- How to choose the <u>optimal</u> weight vector ?
  - Iterative method (Gradient Descent)
  - Closed form solution (OLS)

Gradient Descent Algorithm

Repeat until convergence:

$$w_j = w_j - \eta \frac{\partial J(w)}{\partial w_j} \qquad j = 0,1, \dots, d$$

}

- η = learning rate
- $J(w)$ = cost function $= \frac{1}{2m} \sum_{i=1}^{m} \left(f\left(x^{(i)}\right) - y_i \right)^2$
- $\frac{\partial J(w)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - y_i \right] x_j^{(i)}$
    - d = number of dimensions
    - m = number of training examples
    - $x^{(i)}$ = input variable (features) of the i^{th} training example
    - y_i = output variable of the i^{th} training example
    - $x_j^{(i)}$ = the j^{th} feature in the i^{th} training example

Multivariate Linear Regression

- Linear Regression with multiple variables (features)

Size	No bedrooms	No bathrooms	Price ($1000)
1400	3	2	250
1719	3	2	299
…	…	…	…

- Feature vector and weight vector have d+1 dimensions
- $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \mathrm{w_0x_0} + \mathrm{w_1x_1} + \cdots + \mathrm{w_dx_d}$
- Cost function $= J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}\left(f\left(\mathbf{x}^{(i)}\right) - y_i\right)^2$

Repeat until convergence:

$$w_j = w_j - \eta\,\frac{\partial J(\mathbf{w})}{\partial w_j} \qquad j = 0, \ldots, d$$

}

where $\frac{\partial J(w)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left[f\left(x^{(i)}\right) - y_i\right] x_j^{(i)}$

Stochastic Gradient Descent

- Batch gradient descent uses all training examples for each iteration.
    - Slow, requires large memory but less sensitive to variation
- Stochastic gradient descent updates the parameters with every training example
    - Much faster, allows on-line training but may suffer from high variance

shuffle data set

repeat {

for i = 1, …, m {

$w_j = w_j - \eta \nabla_{w_j} J$ where $\nabla_{w_j} J = \left[f\left(\mathbf{x}^{(i)}\right) - y_i \right] x_j^{(i)}$

$(j = 0, 1, \dots, d)$

}

}

Mini Batch Gradient Descent

- Similar to stochastic gradient descent, but the parameters are updated with n training examples for each iteration
  - Batch Gradient Descent: m training examples per iteration
  - Stochastic Gradient Descent: 1 training example per iteration
  - Mini Batch Gradient Descent: n training examples per iteration, n << m