## Regularized Linear Regression

This is the optimization objective that we came up with last time for regularized linear regression. And we now have the additional regularization term, where lambda ($\lambda$) is our regularization parameter, and we want to find parameters theta that minimizes, this regularized cost function.

$$min_\theta \;\; J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{m} {\theta_j}^2]$$

And we had the following algorithm, for regular linear regression, without regularization, we would repeatedly update the parameters $J(\theta)$ as follows for $(j = 0, 1, 2, 3, \ldots, n)$.

Repeat $\{$

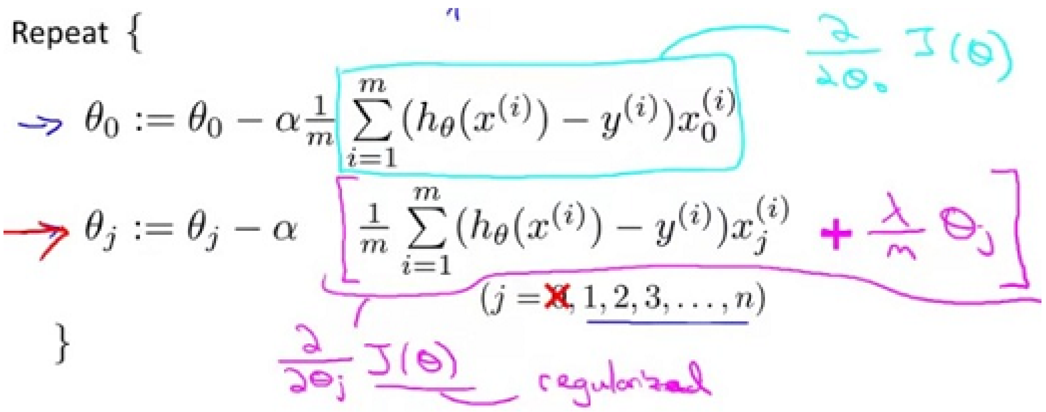
$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$

$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

$(j = 0, 1, 2, 3, \ldots, n)$

$\}$

Concretely, if we want to take this algorithm and modify it to use the regularized objective, all we need to do is take the first term at the bottom and add the following term: $+\frac{\lambda}{m}\theta_j$, and if we implement that, then we have gradient descent for trying to minimize the regularized cost function, $J(\theta)$.



Repeat $\{$

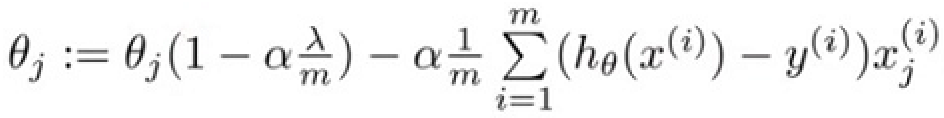
$\theta_0 := \theta_0 - \alpha \frac{1}{m} \underbrace{\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}}_{\frac{\partial}{\partial \theta_0} J(\theta)}$

$\theta_j := \theta_j - \alpha \underbrace{\left[ \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m}\theta_j \right]}_{\frac{\partial}{\partial \theta_j} J(\theta) \text{ regularized}}$

$(j = \cancel{0}, 1, 2, 3, \ldots, n)$

$\}$

So if we group all the terms together that depend on $\theta_j$, we can show that this update can be written equivalently as follows:



$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

**Video Question:** Suppose you are doing gradient descent on a training set of $m > 0$ examples, using a fairly small learning rate $\alpha > 0$ and some regularization parameter $\lambda > 0$. Consider the update rule:

$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

Which of the following statements about the term $(1 - \alpha\frac{\lambda}{m})$ must be true?

- $1 - \alpha\frac{\lambda}{m} > 1$
- $1 - \alpha\frac{\lambda}{m} = 1$

$$1 - \alpha\frac{\lambda}{m} < 1$$

- None of the above.

## Normal Equation

Gradient descent was just one of our two algorithms for fitting a linear regression model. The second algorithm was the one based on the normal equation, where what we did was we created the design matrix $X$ where each row corresponded to a separate training example.

And we created a vector $y$ and that vector contained the labels from our training set. So whereas $X$ is an $m \times (n+1)$ dimensional matrix, $y$ is an $m$ dimensional vector.

$$\underline{X} = \underset{m\times(n+1)}{\begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}}\begin{matrix}\leftarrow \\ \\ \\ \leftarrow\end{matrix} \qquad \underset{\uparrow}{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \mathbb{R}^m$$

And in order to minimize the cost function $J$, we found that one way to do, so is to set $\theta$ with the following formula:

Want $min_\theta \ J(\theta)$

$\theta = (X^TX)^{-1}X^Ty$

And concretely, if you are using regularization, then this formula changes as follows:

$$\theta = \left(X^TX + \lambda \cdot L\right)^{-1}X^Ty$$

$$\text{where } \ L = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

**Non-invertibility (optional/advanced).**

Suppose $m \ (\# \ examples) \leq n \ (\# \ features)$,

$\theta = (X^TX)^{-1}X^Ty$

Now we consider a setting where m, the number of examples, is less than or equal to n, the number of features. If you have fewer examples than features, than this matrix, $X^TX$ will be **non-invertible**, or **singular**. Or the other term for this is the matrix will be **degenerate**.

And if we implement this in Octave anyway and we use the pinv function to take the pseudo inverse, it will kind of do the right thing, but it's not clear that it would give us a very good hypothesis, even though numerically the Octave pinv function will give us a result that kinda makes sense.

If $\lambda > 0$

$X^TX + \lambda L$ will be invertible

Fortunately, regularization also takes care of this for us. And concretely, so long as the regularization parameter lambda is strictly greater than $0$ $(\lambda > 0)$, it is actually possible to prove that the matrix $X^T X + \lambda L$, will not be singular and that this matrix will be invertible. So using regularization also takes care of any non-invertibility issues of the $X^T X$ matrix as well.

## Summary

We can apply regularization to both linear regression and logistic regression. We will approach linear regression first.

### Gradient Descent

We will modify our gradient descent function to separate out $\theta_0$ from the rest of the parameters because we do not want to penalize $\theta_0$

$$\begin{aligned}
& \text{Repeat } \{ \\
& \quad \theta_0 := \theta_0 - \alpha \ \frac{1}{m} \ \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)} \\
& \quad \theta_j := \theta_j - \alpha \ \left[ \left( \frac{1}{m} \ \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \right) + \frac{\lambda}{m}\theta_j \right] \qquad\qquad j \in \{1,2...n\} \\
& \}
\end{aligned}$$

The term $\frac{\lambda}{m}\theta_j$ performs our regularization. With some manipulation our update rule can also be represented as:

$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^m(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$.

The first term in the above equation, $1 - \alpha\frac{\lambda}{m}$ will always be less than $1$. Intuitively you can see it as reducing the value of $\theta_j$ by some amount on every update. Notice that the second term is now exactly the same as it was before.

## Normal Equation

Now let's approach regularization using the alternate method of the non-iterative normal equation. To add in regularization, the equation is the same as our original, except that we add another term inside the parentheses:

$$\begin{aligned}
& \theta = \left( X^TX + \lambda \cdot L \right)^{-1} X^Ty \\
& \text{where } \ L = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}
\end{aligned}$$

$L$ is a matrix with $0$ at the top left and $1$'s down the diagonal, with $0$'s everywhere else. It should have dimension $(n+1) \times (n+1)$. Intuitively, this is the identity matrix (though we are not including $x_0$), multiplied with a single real number $\lambda$.

Recall that if $m < n$, then $X^TX$ is non-invertible. However, when we add the term $\lambda \cdot L$, then $X^TX + \lambda \cdot L$ becomes invertible.