[Summary]Overfitting Problem

Regularization

Regularization is designed to address the problem of overfitting.

High bias or underfitting is when the form of our hypothesis function h maps poorly to the trend of the data. It is usually caused by a function that is too simple or uses too few features. eg. if we take $\mathrm{h}_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ then we are making an initial assumption that a linear model will fit the training data well and will be able to generalize but that may not be the case.

At the other extreme, overfitting or high variance is caused by a hypothesis function that fits the available data but does not generalize well to predict new data. It is usually caused by a complicated function that creates a lot of unnecessary curves and angles unrelated to the data.

This terminology is applied to both linear and logistic regression. There are two main options to address the issue of overfitting:

- **Reduce the number of features**
    a. Manually select which features to keep.
    b. Use a model selection algorithm.
- **Regularization**
    a. Keep all the features, but reduce the parameters

Regularization works well when we have a lot of slightly useful features.

Cost Function

If we have overfitting from our hypothesis function, we can reduce the weight that some of the terms in our function carry by increasing their cost.

Say we wanted to make the following function more quadratic:

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

We'll want to eliminate the influence of $\theta_3 x^3$ and $\theta_4 x^4$. Without actually getting rid of these features or changing the form of our hypothesis, we can instead modify our **cost function**:

$$min\,\theta\,\frac{1}{2m}\sum_{i=1}^{m}\left(\mathrm{h}_\theta\left(x^i\right)-y^i\right)^2+1000\cdot\theta_3^2+1000\cdot\theta_4^2$$

We've added two extra terms at the end to inflate the cost of θ_3 and θ_4. Now, in order for the cost function to get close to zero, we will have to reduce the values of θ_3 and θ_4 to near zero. This will in turn greatly reduce the values of $\theta_3 x^3$ and $\theta_4 x^4$ in our hypothesis function.

We could also regularize all of our theta parameters in a single summation:

$$\min\theta\,\frac{1}{2m}[\sum_{i=1}^{m}\left(\mathrm{h}_\theta\left(x^i\right)-y^i\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2]$$

The λ, or lambda, is the **regularization parameter**. It determines how much the costs of our theta parameters are inflated.

Using the above cost function with the extra summation, we can smooth the output of our hypothesis function to reduce overfitting. If lambda is chosen to be too large, it may smooth out the function too much and cause underfitting.

Regularized Linear Regression

Gradient Descent

We will modify our gradient descent function to separate out θ_0 from the rest of the parameters because we do not want to penalize θ_0.

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\mathrm{h}_\theta\left(x^i\right) - y^i\right) x_0^i$$

$$\theta_1 := \theta_j - \alpha[(\frac{1}{m} \sum_{i=1}^{m} \left(\mathrm{h}_\theta\left(x^i\right) - y^i\right) x_j^i) + \frac{\lambda}{m}\theta_j] \quad j \in \{1,2 \ldots n\}$$

}

The term $\frac{\lambda}{m}\theta_j$ performs our regularization.

With some manipulation our update rule can also be represented as:

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\mathrm{h}_\theta\left(x^i\right) - y^i\right) x_j^i$$

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 θ_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:

$$\theta = (X^T X + \lambda \cdot L)^{-1} X^T y$$

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

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)×(n+1). Intuitively, this is the identity matrix (though we are not including θ_0), multiplied with a single real number λ.

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

Regularized Logistic Regression

We can regularize logistic regression in a similar way that we regularize linear regression.

Cost Function

Recall that our cost function for logistic regression was:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \log\left(\mathrm{h}_\theta\left(x^i\right)\right) + \left(1 - y^i\right)\log\left(1 - \mathrm{h}_\theta\left(x^i\right)\right)\right]$$

We can regularize this equation by adding a term to the end:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \log\left(\mathrm{h}_\theta(x^i)\right) + (1 - y^i)\log\left(1 - \mathrm{h}_\theta(x^i)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Note Well: The second sum, $\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ **means to explicitly exclude** the bias term, θ_0. I.e. the θ vector is indexed from 0 to n (holding n+1 values, θ_0 through θ_n), and this sum explicitly skips θ_0, by running from 1 to n, skipping 0.

Gradient Descent

Just like with linear regression, we will want to **separately** update θ_0 and the rest of the parameters because we do not want to regularize θ_0.

Repeat {

$$\theta_0 \coloneqq \theta_0 - \frac{\alpha}{m}\sum_{i=1}^{m}(\mathrm{h}_\theta(x^i) - y^i)x_0^i$$

$$\theta_j \coloneqq \theta_j - \alpha[(\frac{1}{m}\sum_{i=1}^{m}(\mathrm{h}_\theta(x^i) - y^i)x_j^i) + \frac{\lambda}{m}\theta_j] \quad j \in \{1,2 \ldots n\}$$

}

This is identical to the gradient descent function presented for linear regression.