Lecture 5

Regularization

Dr. Le Huu Ton

Hanoi, 2018

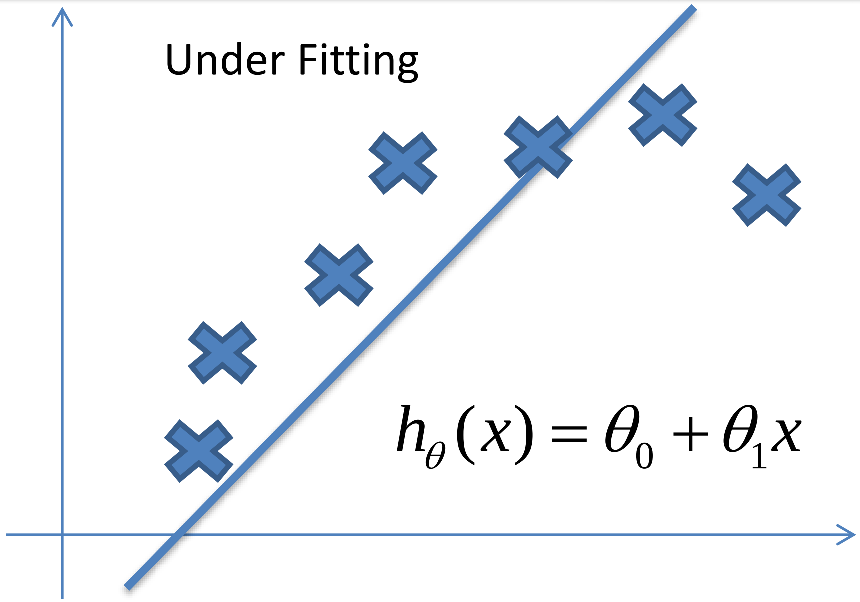
Outline

- Overfitting Problem
- Regularization
- Regularization with Linear Regression
- Regularization with Logistic Regression

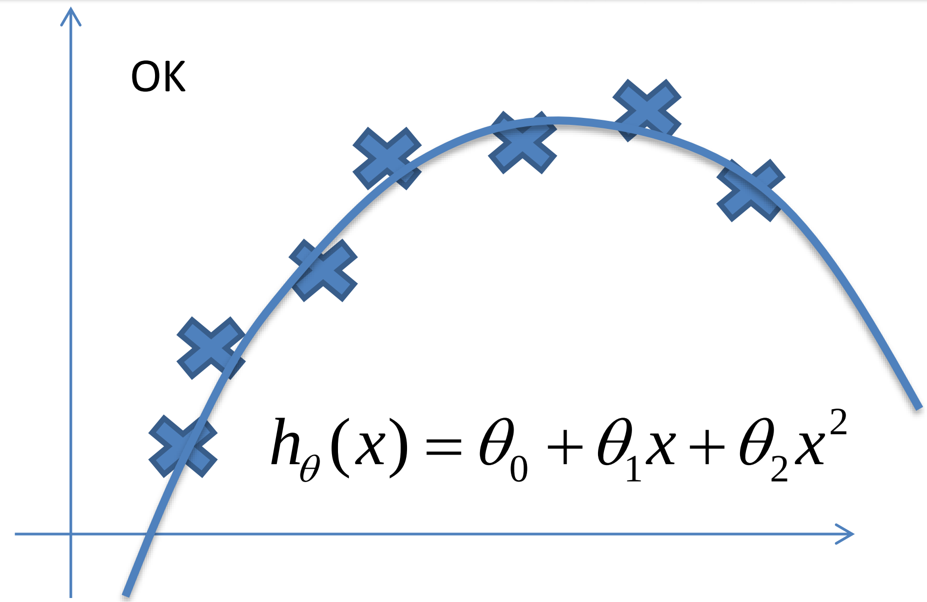
Outline

- Overfitting Problem
- Regularization
- Regularization with Linear Regression
- Regularization with Logistic Regression

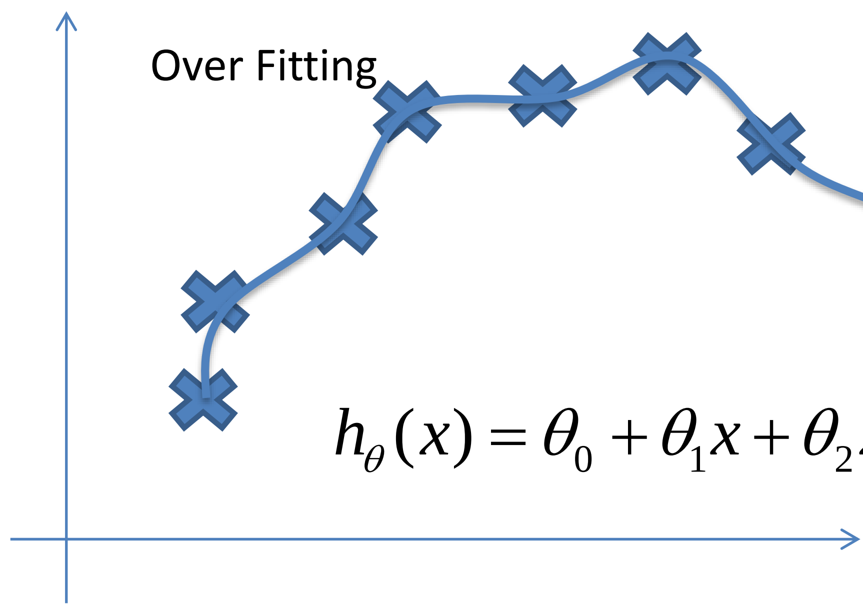
Overfitting Problem



$$h_\theta(x) = \theta_0 + \theta_1 x$$

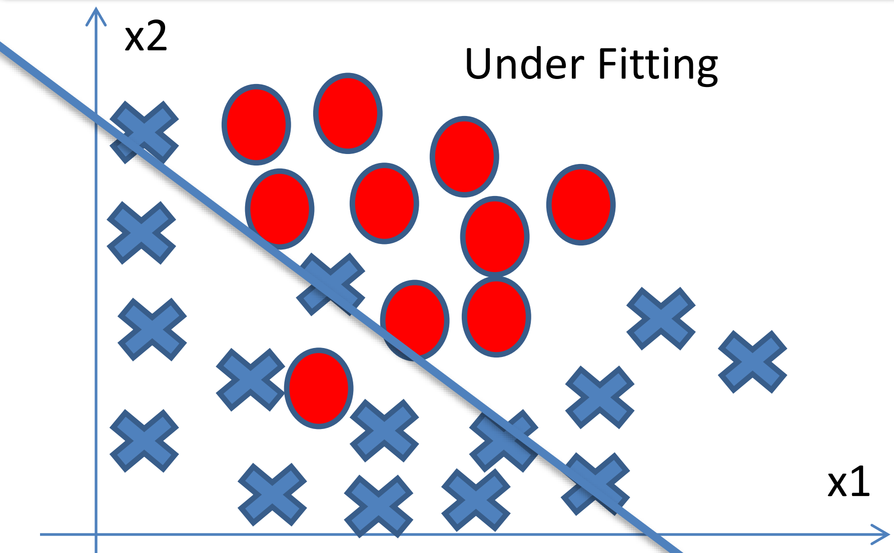


$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

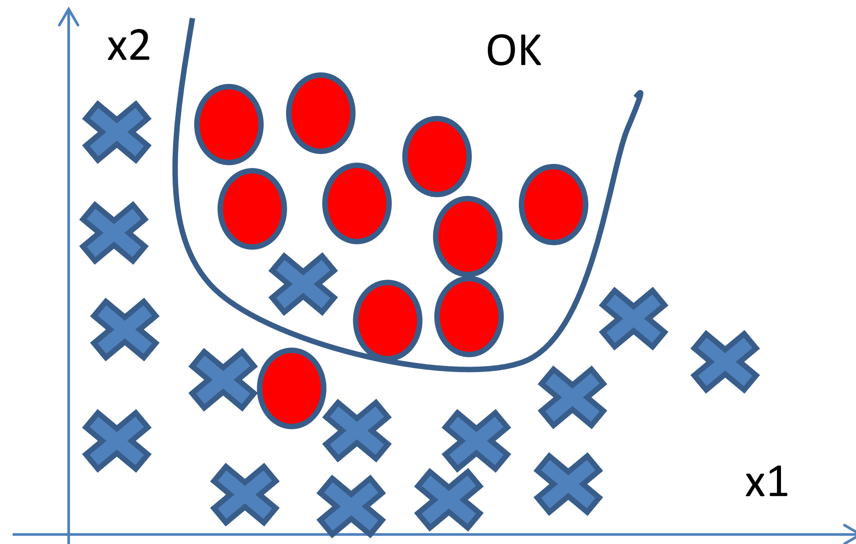


$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \ldots$$

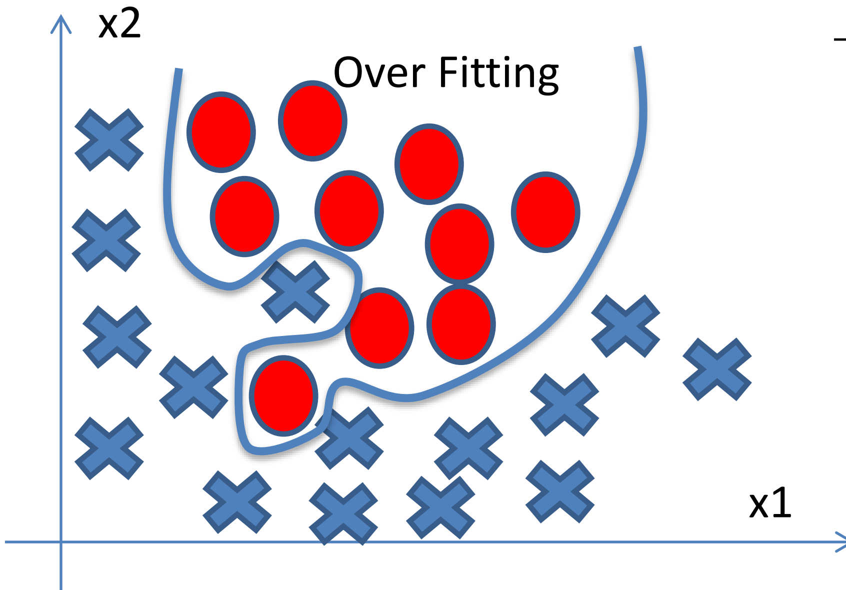
Overfitting Problem



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1^3 + \theta_6 x_2^3 + \ldots)$$

Overfitting Problem

Under fitting:

Under fitting refers to a model that can neither model the training data not generalize to new data.

An under fit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

Over Fitting :

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance on the model on new data.

Outline

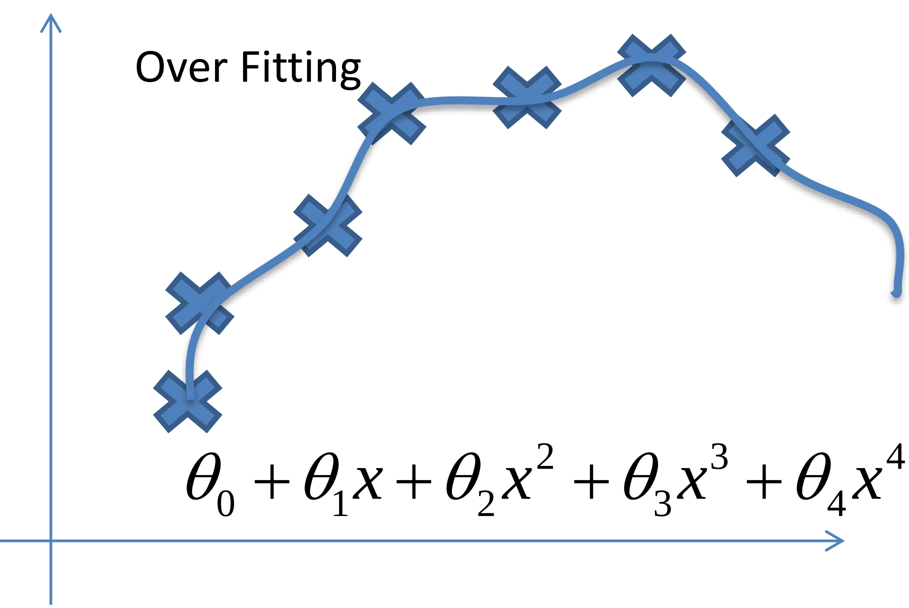
- Overfitting Problem
- Regularization
- Regularization with Linear Regression
- Regularization with Logistic Regression

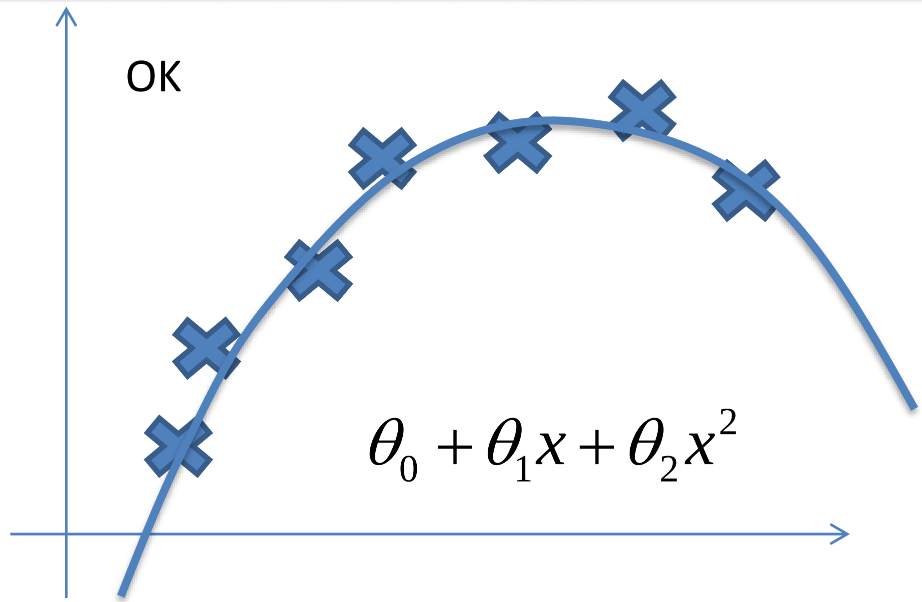
Regularization

Regularization is a *technique* used in an attempt to solve the **overfitting** problem.

Regularization is done by reduce the magnitude of some coefficient θ_j

Overfitting Problem





Regularization: reduce value of θ_3 and θ_4

Minimize the cost function

$$E(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \quad +9999\theta_3 + 9999\theta_4$$

$$=> \theta_3 \approx 0, + \theta_4 \approx 0$$

Regularization

Small values of coefficients $\theta_0, \theta_1, \ldots \theta_n$

$\Rightarrow$ Simpler hypothesis h(x)

$\Rightarrow$ Less prone to overfitting

<u>Regularization</u>: Add a regularization component into the cost function

$$E(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

Regularization component

Regularization

Question:

What if λ is set by a extremely large number (too large for our problem), which of the following statement is correct:

1. The algorithm works fine
2. Algorithm fail to eliminate overfitting
3. Algorithm results in under fitting
4. Gradient descent will fail to converge

Outline

- Overfitting Problem
- Regularization
- Regularization with Linear Regression
- Regularization with Logistic Regression

Regularization with Linear Regression

Regularization:

Minimize the Cost Function

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

Gradient descent:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial_{\theta_j}} E(\theta)$$

Regularization with Linear Regression

Gradient Descent:

Repeat until converged:

{

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

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x_j^i + \frac{\alpha\lambda}{m} \theta_j \forall j = 1 : n$$

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x_j^i \quad \forall j = 1 : n$$

}

Regularization with Linear Regression

Normal Equation without regularization:

$$\theta = (X^T X)^{-1} X^T Y$$

Normal Equation with regularization

$$\theta = (X^T X + \lambda \begin{vmatrix} 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & 0 \\ \ldots & \ldots & 1 & \ldots \\ 0 & 0 & 0 & 1 \end{vmatrix})^{-1} X^T Y$$

Outline

- Overfitting Problem
- Regularization
- Regularization with Linear Regression
- Regularization with Logistic Regression

Regularization with Logistic Regression

***Logistic Regression*: Minimize the cost function**

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

Gradient descent:

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial_{\theta_j}}E(\theta)$$

Regularization with Logistic Regression

Gradient Descent:

Repeat until converged:

{

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

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} [(h(x^{(i)}) - y^{(i)}) x_0^i] - \frac{\lambda}{m} \theta_j \forall j = 1 : n$$

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x_0^i \quad \forall j = 1 : n$$

}

Regularization with Logistic Regression

Newton's Method with Regularization

$$\theta^{t+1} := \theta^{t} - H^{-1}\Delta_{\theta}E$$

$$\Delta_{\theta}E = \begin{vmatrix} \frac{\partial}{\partial_{\theta_0}} E(\theta) \\ \ldots \\ \frac{\partial}{\partial_{\theta_n}} E(\theta) \end{vmatrix} = \begin{vmatrix} \frac{1}{m}\sum (h(x^{(i)}) - y^{(i)})x_0^i \\ \frac{1}{m}\sum (h(x^{(i)}) - y^{(i)})x_1^i - \frac{\lambda}{m}\theta_1 \\ \ldots \\ \frac{1}{m}\sum (h(x^{(i)}) - y^{(i)})x_n^i - \frac{\lambda}{m}\theta_n \end{vmatrix}$$

Regularization with Logistic Regression

Hessian Matric:

$$H = \frac{1}{m}\sum_{i=1}^{m}\left[h(x^{(i)})(1-h(x^{(i)})x^{(i)}(x^{(i)})^{T}\right] + \lambda\begin{vmatrix} 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & 0 \\ \ldots & \ldots & 1 & \ldots \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Regularization with Logistic Regression

When using regularized logistic regression, which of these is the best way to monitor whether gradient descent is working correctly?

- ○ Plot $-\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$ as a function of the number of iterations, and make sure it's decreasing.
- ○ Plot $-\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right] - \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations, and make sure it's decreasing.
- ○ Plot $-\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations, and make sure it's decreasing.
- ○ Plot $\sum_{j=1}^{n}\theta_j^2$ as a function of the number of iterations, and make sure it's decreasing.

References

http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning

Andrew Ng Slides:

https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&sqi=2&ved=0ahUKEwjNt4fdvMDPAhXIn5QKHZO1BSgQFggfMAE&url=https%3A%2F%2Fdatajobs.com%2Fdata-science-repo%2FGeneralized-Linear-Models-%5BAndrew-Ng%5D.pdf&usg=AFQjCNGq37q2uVFcpGhNqH-5KZSlJ_HSxg&sig2=vnCEvyvKQGCuryttAPcokw&bvm=bv.134495766,d.dGo