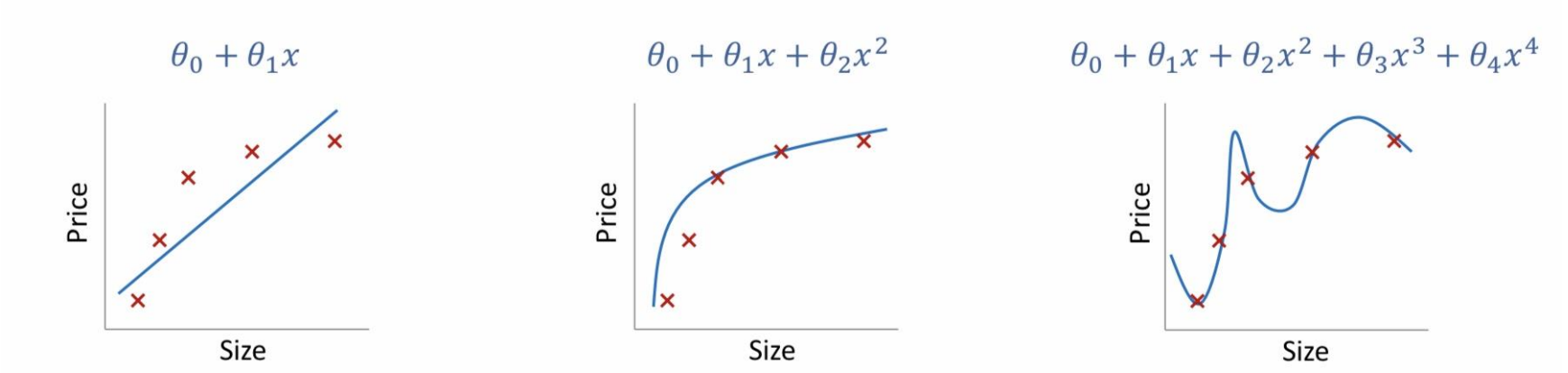
Machine Learning

By Ghazal Lalooha

Regularization: facing with over-fit

Example: Polynomial regression



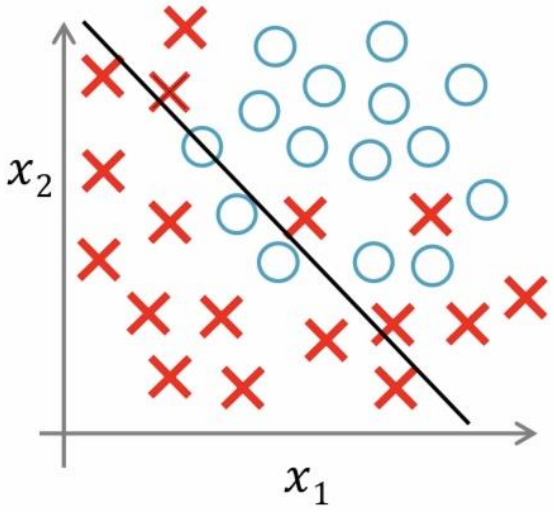
Under-fit (high bias)

Correct model

Over-fit (high variance)

Overfitting: If the number of features is very large, the learned hypothesis may learn the training data very well, but it is also possible that the hypothesis will fail to predict new data. (lack of generalizability)

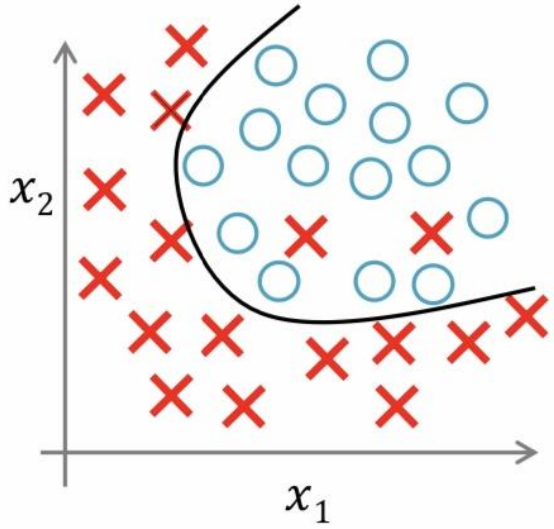
Example: Logistic Regression



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

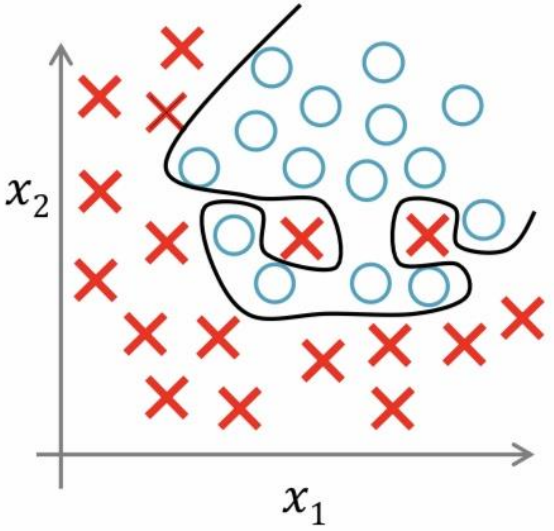
(g = sigmoid function)

Under-fit (high bias)



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$

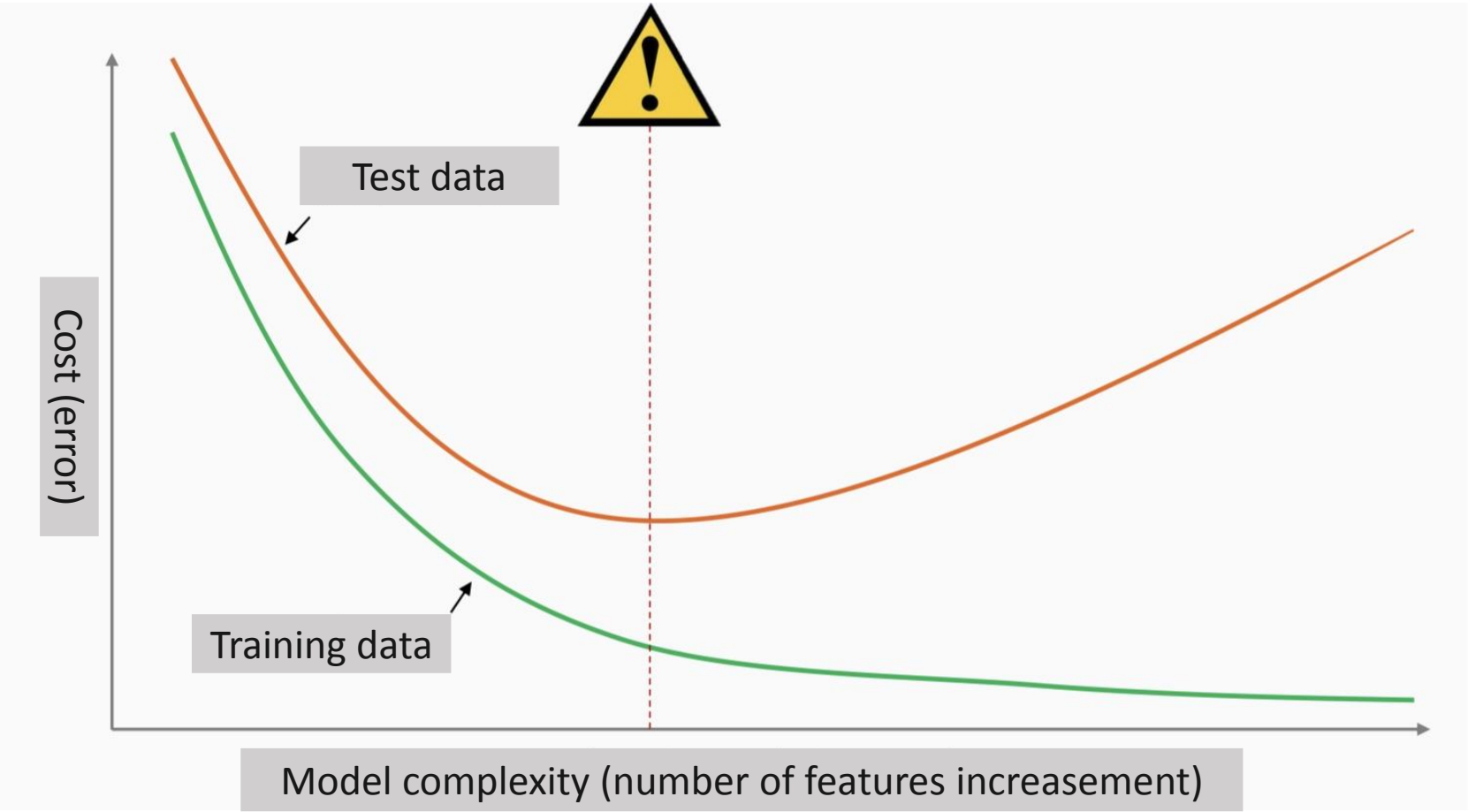
Correct model



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2$

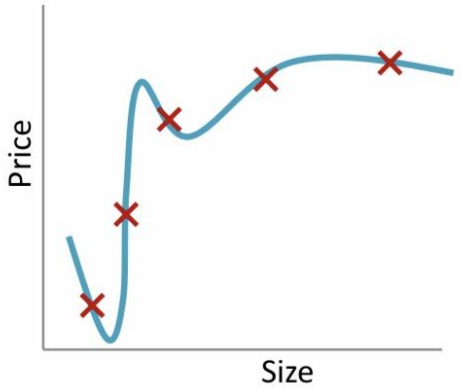
Over-fit (high variance)

Over-fit


Test data
Training data
Cost (error)
Model complexity (number of features increasement)

Facing with Over-fit

- Features
- X_1: house size
- X_2: number of bedrooms
- X_3: number of levels
- X_4: age
- X_5: kitchen size
- X_6: number of bathrooms
- …
- X_{100}: average income of neighbors

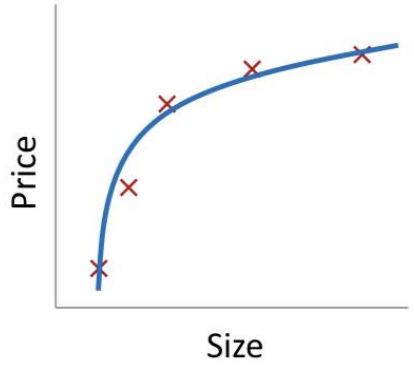


Facing with Over-fit

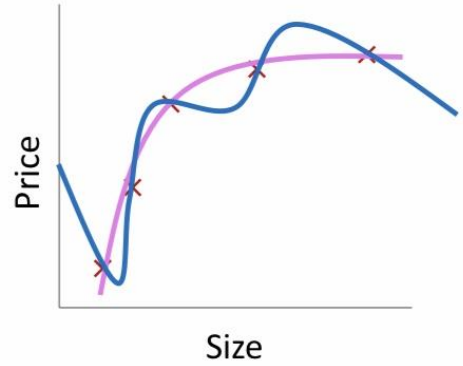
- Possible solutions:
  - Reduce the number of features:
    - Manually select the most important features and delete the rest.
    - Algorithms for model selection (to be continued)
  - Regularization:
    - Keep all features, but reduce the value of θ_j parameters.
    - This method works well when we have many features, each of which has a small contribution in predicting the output value.

Cost Function

Regularization Concept



$\theta_0 + \theta_1 x + \theta_2 x^2$



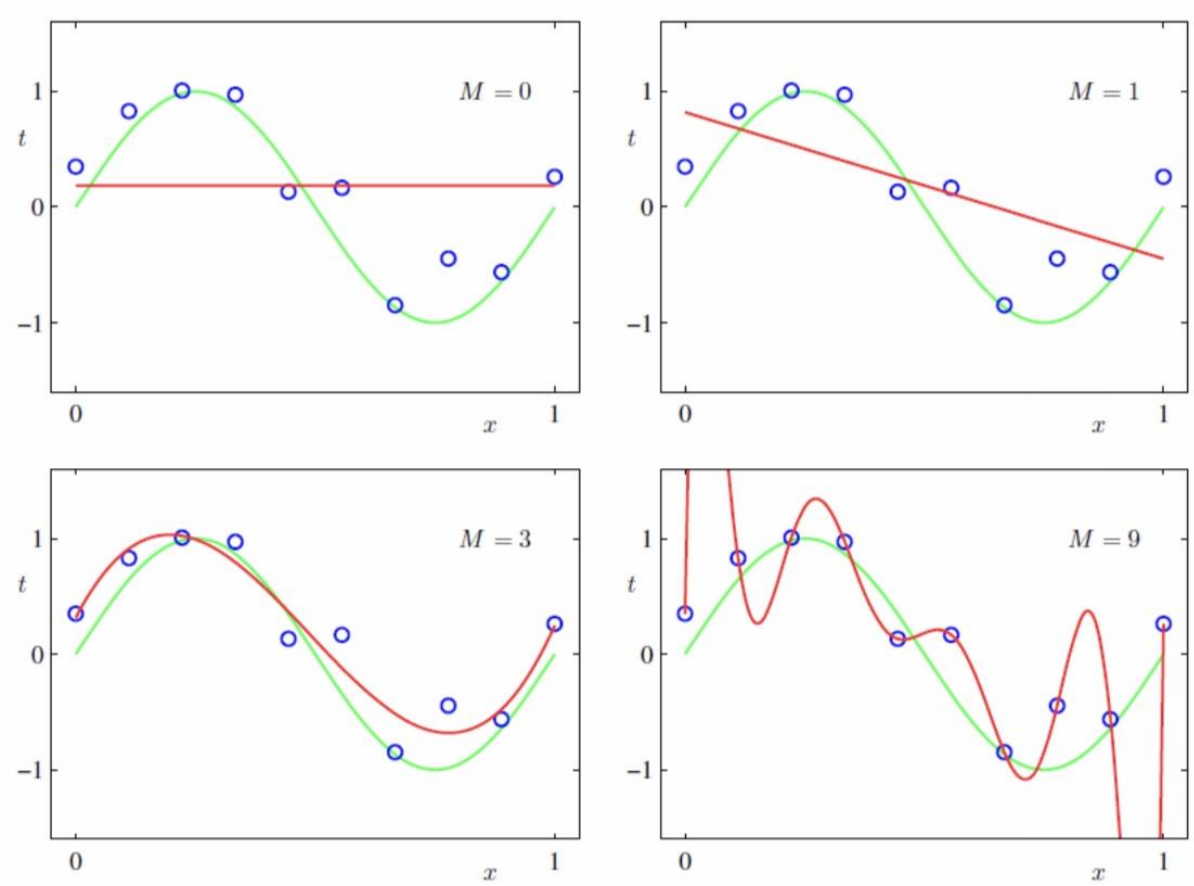
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Cost Function:

By penalizing the cost function, the values of θ_3 and θ_4 parameters can be very small:

$$\min_\theta \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \underbrace{1000\theta_3^2}_{\approx 0} + \underbrace{1000\theta_4^2}_{\approx 0}$$

Regularization Concept



	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

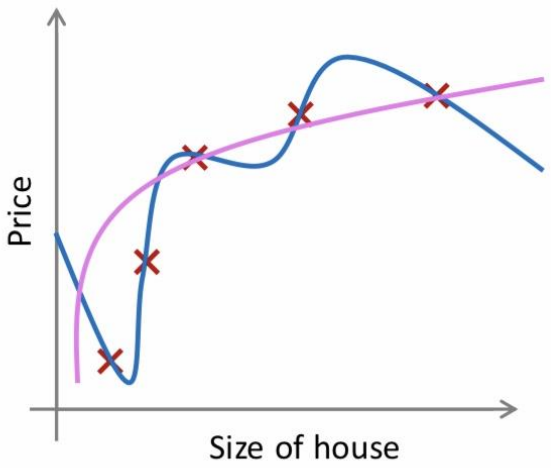
Regularization

- Regularization: use small values for theta parameters
  - Creating "simpler" hypotheses
  - "Occam's principle of shaving" shaving unnecessary parts from the model
  - Reduce the risk of overfitting
- Example:
  - Features: $x_1, x_2, \ldots, x_{100}$
  - Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

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

Regularization

- Regularization



Regularization factor

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

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

Class Question

- In regularized linear regression, the parameter values are chosen in such a way that the value of the cost function is minimized.

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

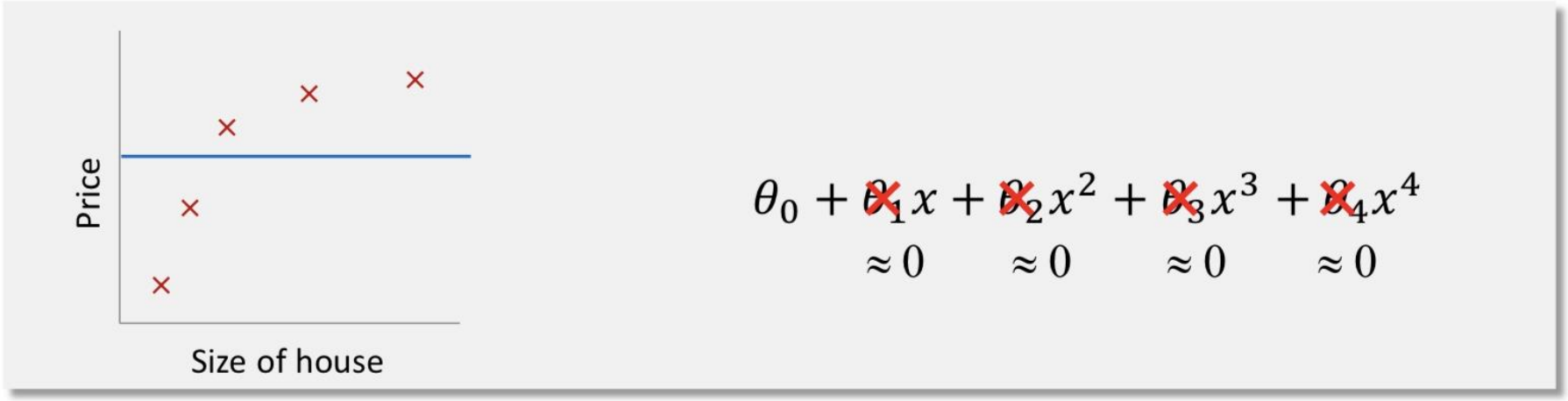
- Question: If we set the lambda regularization coefficient with a very large value (for example 10^{10}), what will happen in this case?
  - The algorithm works well and a large value of lambda does not harm it.
  - The algorithm fails when dealing with overfitting.
  - The algorithm ends with a under-fitting.
  - The gradient descent algorithm does not converge.

Class Question

In regularized linear regression, the parameter values are chosen in such a way that the value of the cost function is minimized.

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

Question: If we set the lambda regularization coefficient with a very large value (for example 10^{10}), what will happen in this case?



Regularized Linear Regression

Regularized Linear Regression

- Cost function

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

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

Gradient Descent (without regularization)

- Without using regularization

repeat until convergence {

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad (j = 0,1,2, \dots, n)$$

}

Gradient Descent (with regularization)

- Using regularization

repeat until convergence {

$$\theta_0 = \theta_0 - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_0^{(i)}$$

$$\theta_j = \theta_j - \alpha \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} + \lambda\theta_j\right] \qquad (j = 1,2,\dots,n)$$

}

$$\theta_j = \theta_j \underbrace{(1 - \alpha\lambda)}_{<1} - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)}$$

Normal Equation

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$$X\,\theta = y$$

Not using regularization for θ_0 parameter

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \right)^{-1} X^T y$$

Irreversibility

Suppose m < n:

$$\theta = \underbrace{(X^T X)^{-1}}_{\text{Irreversibility}} X^T y$$

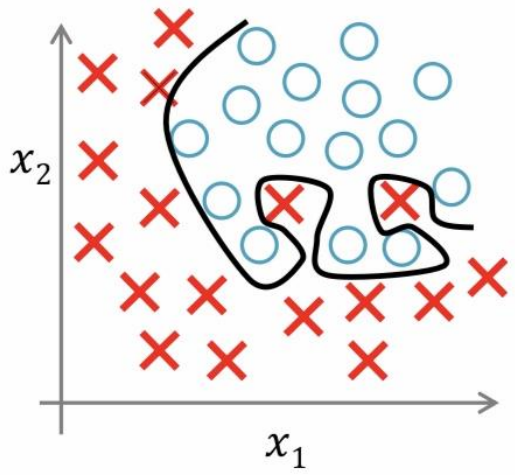
If $\lambda > 0$:

$$\theta = \overbrace{\left(X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \right)^{-1}}^{\text{reversibility}} X^T y$$

Regularized Logistic Regression

Logistic Regression

Hypothesis



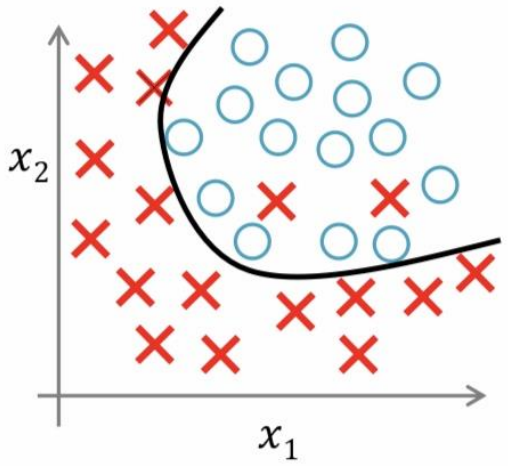
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \cdots)$$

Cost function

$$J(\theta) = -\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

Logistic Regression

Hypothesis



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

Cost function

$$J(\theta) = -\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right) + \frac{\lambda}{2} \sum_{i=1}^{m} \theta_j^2$$

Gradient Descent

- Using regularization

repeat until convergence {

$$\theta_0 = \theta_0 - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_0^{(i)}$$

$$\theta_j = \theta_j - \alpha \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} + \lambda\theta_j\right] \qquad (j = 1,2,\dots,n)$$

}

$$h_\theta\left(x^{(i)}\right) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$

Advanced Optimization

```
function [jVal, gradient] = costFunction(theta)
```

jVal = [code to compute $J(\theta)$];

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

gradient(1) = [code to compute $\frac{\partial}{\partial \theta_0} J(\theta)$];

$$\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_0^{(i)}$$

gradient(2) = [code to compute $\frac{\partial}{\partial \theta_1} J(\theta)$];

$$\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_1^{(i)} + \lambda \theta_1$$

...

gradient(n+1) = [code to compute $\frac{\partial}{\partial \theta_n} J(\theta)$];