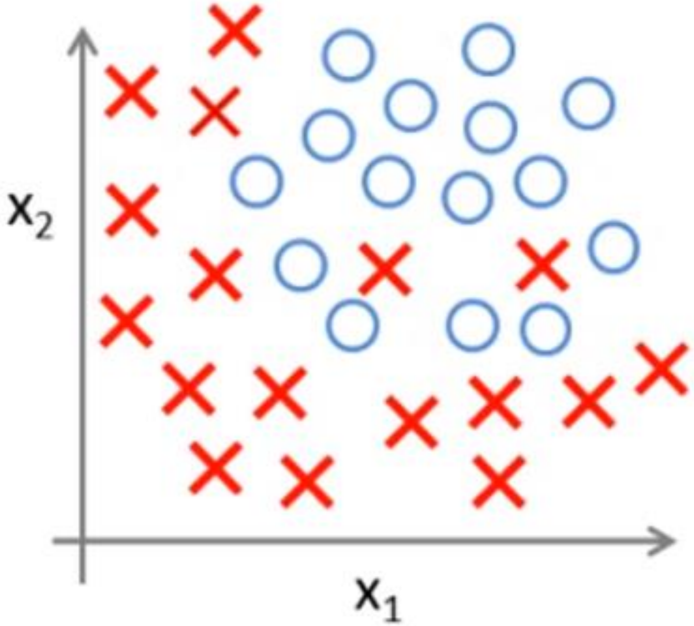
Regularized Logistic Regression

Solving the Problem of Overfitting

Regularization

Regularized logistic regression.

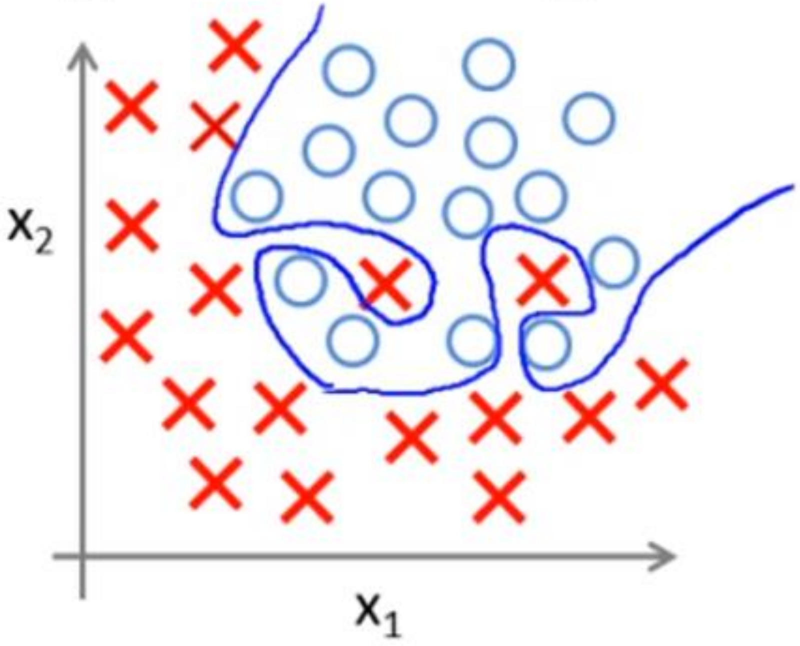


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

Cost function:

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

Regularized logistic regression.



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

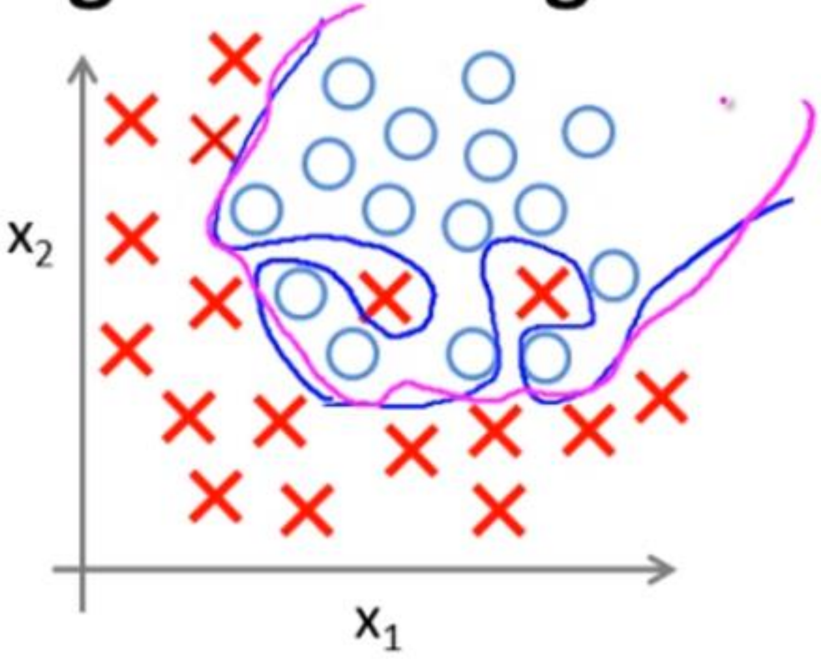
Cost function:

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

$$+ \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$\theta_1, \theta_2, \dots \theta_n$

Regularized logistic regression.



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

Cost function:

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

$$+ \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \quad \Big| \; \boxed{\theta_1, \theta_2, \dots, \theta_n}$$

Gradient descent

Repeat $\{$

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

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

$\}$

Alt

Gradient descent

Repeat $\{$

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

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

$$(j = \cancel{0}, \underbrace{1, 2, 3, \ldots, n}_{\theta_1 \ldots\ldots \theta_n})$$

$\}$

Gradient descent

Repeat $\{$

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

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

$$(j = \cancel{0}, \underbrace{1, 2, 3, \dots, n}_{\theta_1 \dots \theta_n})$$

$\}$

Gradient descent

Repeat $\{$

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

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

$$(j = \cancel{0}, \underbrace{1, 2, 3, \dots, n}_{\theta_1 \dots \theta_n})$$

$\}$

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

Exercise

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

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

Advanced optimization

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

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

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

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

`gradient(3) = [` code to compute $\frac{\partial}{\partial\theta_2}J(\theta)$ `];`

$\vdots$

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

Advanced optimization

$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \vdots \\ \Theta_n \end{bmatrix}$

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

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

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

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

gradient(3) = [code to compute $\frac{\partial}{\partial\theta_2}J(\theta)$];

$\vdots$

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

$$\begin{array}{cl} \Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \vdots \\ \Theta_n \end{bmatrix} & \begin{array}{l} \text{theta(1)} \\ \text{theta(2)} \\ \\ \text{theta(n+1)} \end{array} \end{array}$$

Advanced optimization

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

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

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

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

gradient(3) = [code to compute $\frac{\partial}{\partial\theta_2}J(\theta)$] ;

$\vdots$

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

Advanced optimization

fminunc(@costFunction...)

$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \vdots \\ \Theta_n \end{bmatrix}$ theta(1) theta(2) theta(n+1)

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

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

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

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

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

gradient(3) = [code to compute $\frac{\partial}{\partial \theta_2} J(\theta)$] ;

⋮

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

Advanced optimization

fminunc (@costFunction...)

$\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$ theta(1), theta(2), theta(n+1)

```
function [jVal, gradient] = costFunction(theta)
    jVal = [code to compute J(θ)];
```

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

```
    gradient(1) = [code to compute ∂/∂θ0 J(θ)];
```

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

```
    gradient(2) = [code to compute ∂/∂θ1 J(θ)];
```

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

```
    gradient(3) = [code to compute ∂/∂θ2 J(θ)];
```

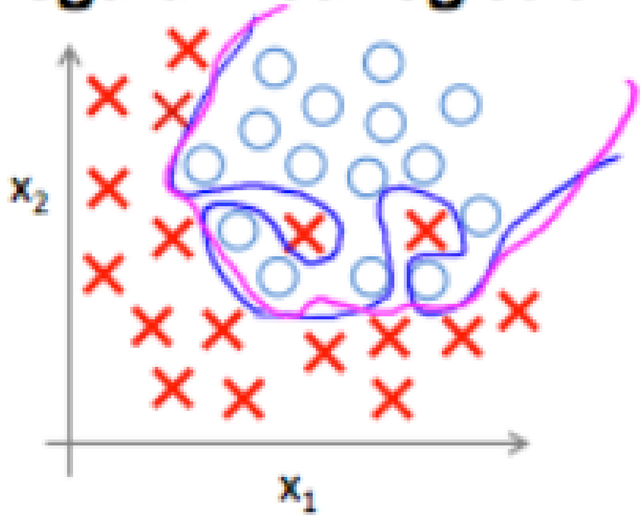
$$\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}\right) + \frac{\lambda}{m}\theta_2$$

⋮

```
    gradient(n+1) = [code to compute ∂/∂θn J(θ)];
```

Summary

Regularized logistic regression.



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

Summary

- Recall that our cost function for logistic regression was:
- $$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1-y^{(i)})\ \log(1-h_\theta(x^{(i)}))]$$

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

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

Summary

Gradient descent

Repeat {

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

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

$$(j = \cancel{0}, \underbrace{1, 2, 3, \ldots, n}_{\theta_1 \ldots \theta_n})$$

}

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